DynamoDB Hands-On Lab Runbook

Introduction

Goals: This runbook guides you through the process of setting up and managing Amazon DynamoDB using AWS CloudShell. The goals of this simulation are:

1. To create DynamoDB tables.
2. To load sample data into the tables.
3. To enable Point-In-Time Recovery (PITR) backups.
4. To simulate on-demand backups.

What You Will Accomplish

By the end of this simulation, you will:

- Have DynamoDB tables populated with sample data.
- Understand how to enable and manage backups for DynamoDB tables.
- Use AWS CloudShell for running AWS CLI commands to interact with DynamoDB.

Using AWS CloudShell

AWS CloudShell is a browser-based shell environment that allows you to run AWS CLI commands directly from the AWS Management Console. Unlike AWS Cloud9, which is an integrated development environment, CloudShell is ideal for quick command-line tasks without needing to set up a development environment. Using the CLI for steps 1-6 provides automation, repeatability, and ease of execution, especially when dealing with multiple resources.

Scenario: E-Commerce Platform

You are developing an e-commerce platform where users can browse a catalog of products, participate in forums, discuss specific threads, and post replies. Each type of interaction requires structured data storage, and DynamoDB offers a scalable solution for handling these requirements.

Table Breakdown and Explanation:

1. **ProductCatalog table**: Stores details about the products available for purchase.Each product has a unique ID and attributes such as name, price, and category.Used to display products on the e-commerce platform. Allows customers to view product details and make purchases.
2. **Forum table**: Manages the forums where users can create and participate in discussions. Each forum has a unique name.
3. **Thread table**: Represents individual discussion topics within forums.

4. **Reply table**:Stores replies to threads.Each reply has a unique ID and a timestamp to indicate when it was posted.

Steps to Follow

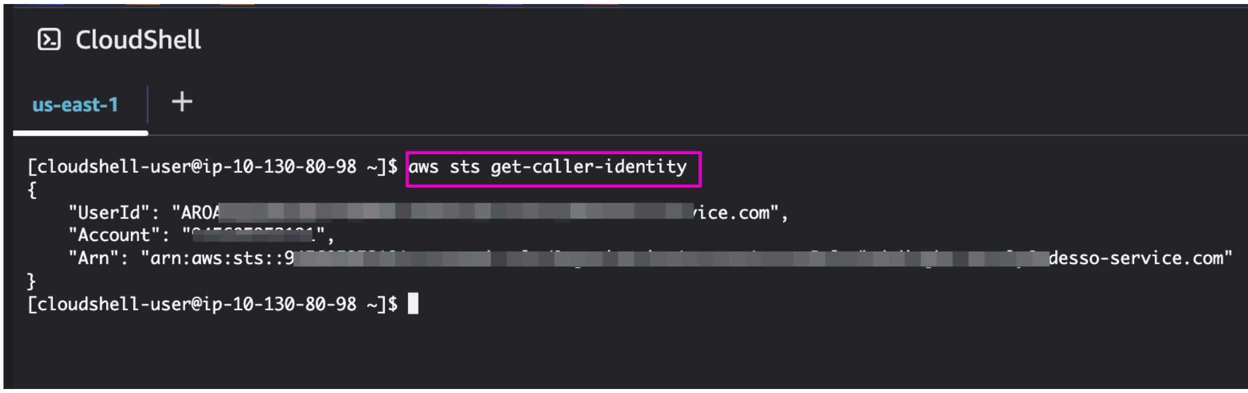
Step 1: Open AWS CloudShell

- In the AWS Management Console, search and click on the **CloudShell** icon
- CloudShell will open in a new browser tab.

Step 2: Verify AWS Credentials

- In the CloudShell terminal, run the following command to verify your credentials:

$ aws sts get-caller-identity



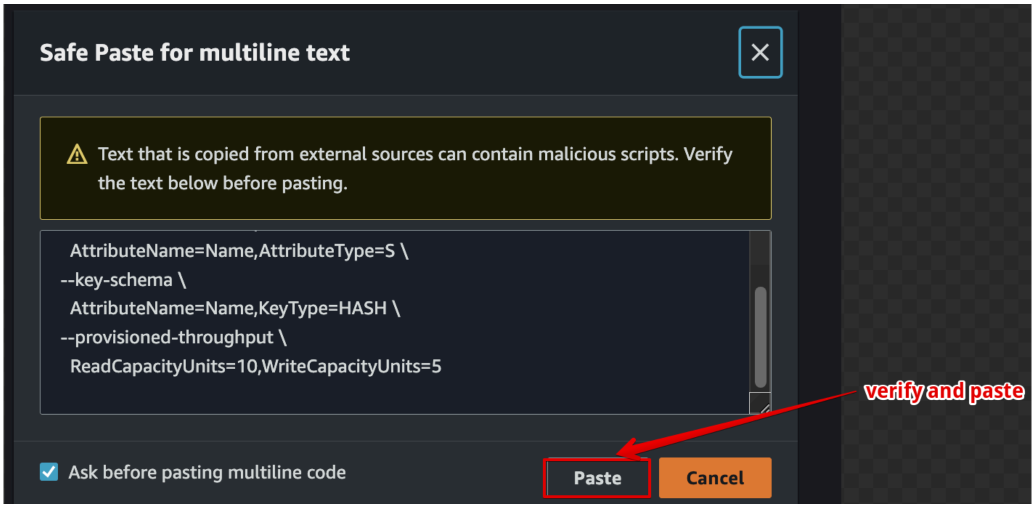
```
[cloudshell-user@ip-10-130-80-98 ~]$ aws sts get-caller-identity
{
    "UserId": "AROA[redacted]ice.com",
    "Account": "[redacted]",
    "Arn": "arn:aws:sts::9[redacted]desso-service.com"
}
[cloudshell-user@ip-10-130-80-98 ~]$
```

- This command outputs details of the IAM user or role that you are using in CloudShell as shown above. Ensure the user or role has the necessary permissions to create and manage DynamoDB tables and perform backup operations.

Step 3: Create DynamoDB Tables

- In CloudShell, create the ProductCatalog table using the AWS CLI:

```
aws dynamodb create-table \
  --table-name ProductCatalog \
  --attribute-definitions \
    AttributeName=Id,AttributeType=N \
  --key-schema \
    AttributeName=Id,KeyType=HASH \
  --provisioned-throughput \
    ReadCapacityUnits=10,WriteCapacityUnits=5
```

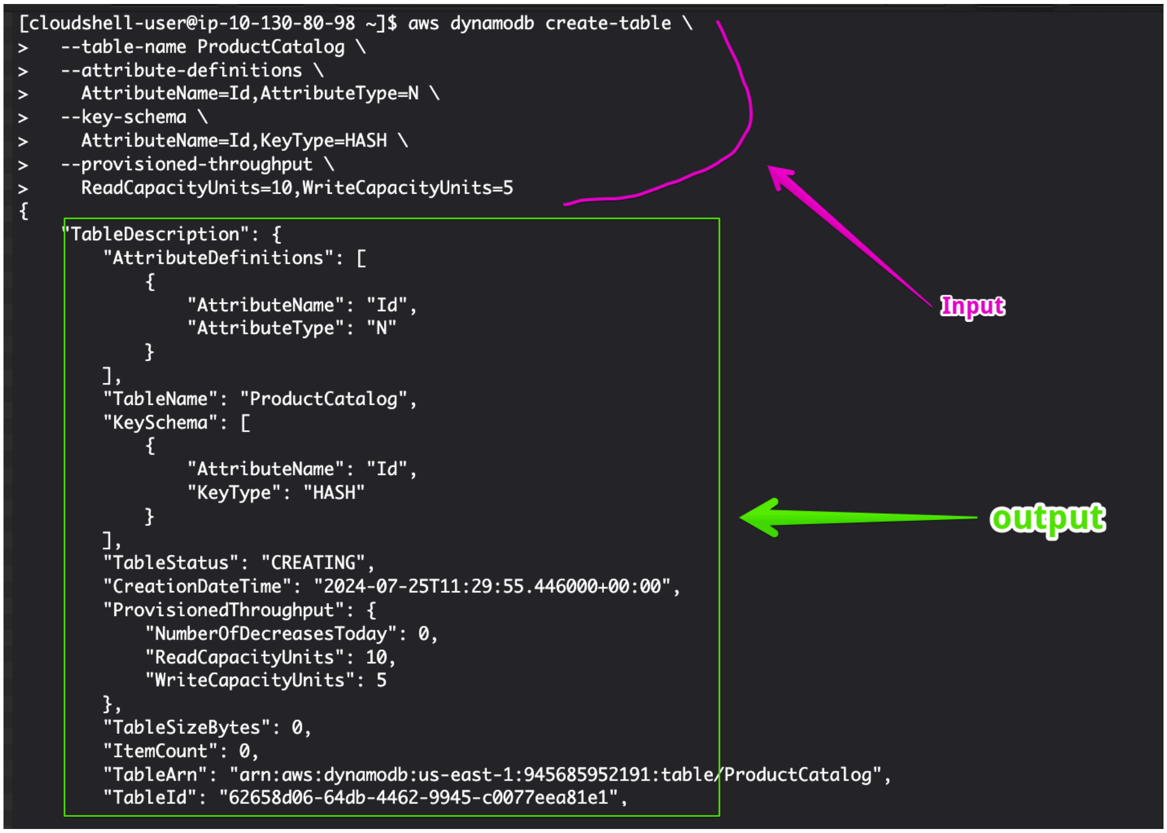


Safe Paste for multiline text

⚠ Text that is copied from external sources can contain malicious scripts. Verify the text below before pasting.

```
  AttributeName=Name,AttributeType=S \
--key-schema \
  AttributeName=Name,KeyType=HASH \
--provisioned-throughput \
  ReadCapacityUnits=10,WriteCapacityUnits=5
```

☑ Ask before pasting multiline code



```
[cloudshell-user@ip-10-130-80-98 ~]$ aws dynamodb create-table \
>   --table-name ProductCatalog \
>   --attribute-definitions \
>     AttributeName=Id,AttributeType=N \
>   --key-schema \
>     AttributeName=Id,KeyType=HASH \
>   --provisioned-throughput \
>     ReadCapacityUnits=10,WriteCapacityUnits=5
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Id",
                "AttributeType": "N"
            }
        ],
        "TableName": "ProductCatalog",
        "KeySchema": [
            {
                "AttributeName": "Id",
                "KeyType": "HASH"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2024-07-25T11:29:55.446000+00:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-east-1:945685952191:table/ProductCatalog",
        "TableId": "62658d06-64db-4462-9945-c0077eea81e1",
```

2. **Create the Forum Table**:
    - Create the Forum table using the AWS CLI:

      ```
      aws dynamodb create-table \
        --table-name Forum \
        --attribute-definitions \
          AttributeName=Name,AttributeType=S \
        --key-schema \
          AttributeName=Name,KeyType=HASH \
        --provisioned-throughput \
          ReadCapacityUnits=10,WriteCapacityUnits=5
      ```

3. **Create the Thread Table**:
    - Create the Thread table using the AWS CLI:

      ```
      aws dynamodb create-table \
        --table-name Thread \
        --attribute-definitions \
          AttributeName=ForumName,AttributeType=S \
          AttributeName=Subject,AttributeType=S \
        --key-schema \
          AttributeName=ForumName,KeyType=HASH \
          AttributeName=Subject,KeyType=RANGE \
        --provisioned-throughput \
          ReadCapacityUnits=10,WriteCapacityUnits=5
      ```

4. **Create the Reply Table**:
    - Create the Reply table using the AWS CLI:

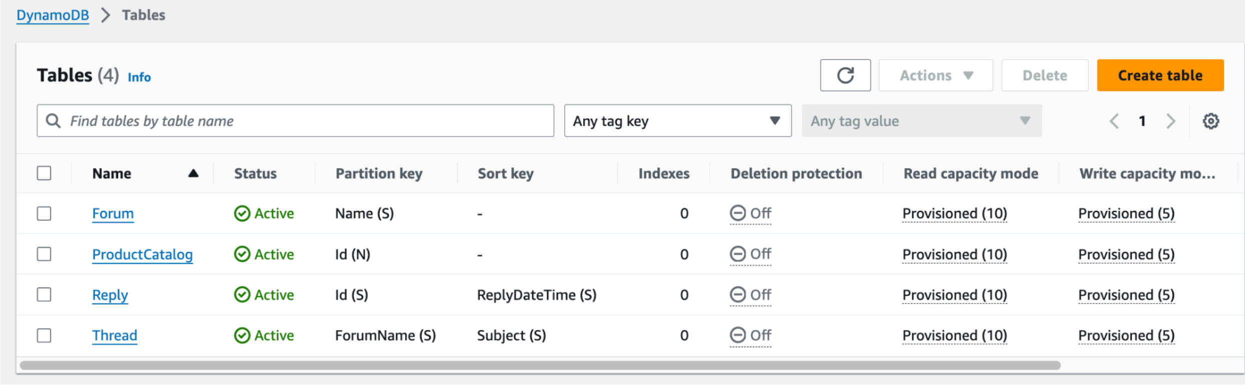
      ```
      aws dynamodb create-table \
        --table-name Reply \
        --attribute-definitions \
          AttributeName=Id,AttributeType=S \
          AttributeName=ReplyDateTime,AttributeType=S \
        --key-schema \
          AttributeName=Id,KeyType=HASH \
          AttributeName=ReplyDateTime,KeyType=RANGE \
        --provisioned-throughput \
          ReadCapacityUnits=10,WriteCapacityUnits=5
      ```

5. **Wait for Tables to be Created**:
    - Ensure all tables are created before proceeding:

      ```
      aws dynamodb wait table-exists --table-name ProductCatalog && \
      aws dynamodb wait table-exists --table-name Reply && \
      aws dynamodb wait table-exists --table-name Forum && \
      aws dynamodb wait table-exists --table-name Thread
      ```

6. **Verify Table Creation**:
   - Confirm that all 4 tables have been created but are currently empty with no items:

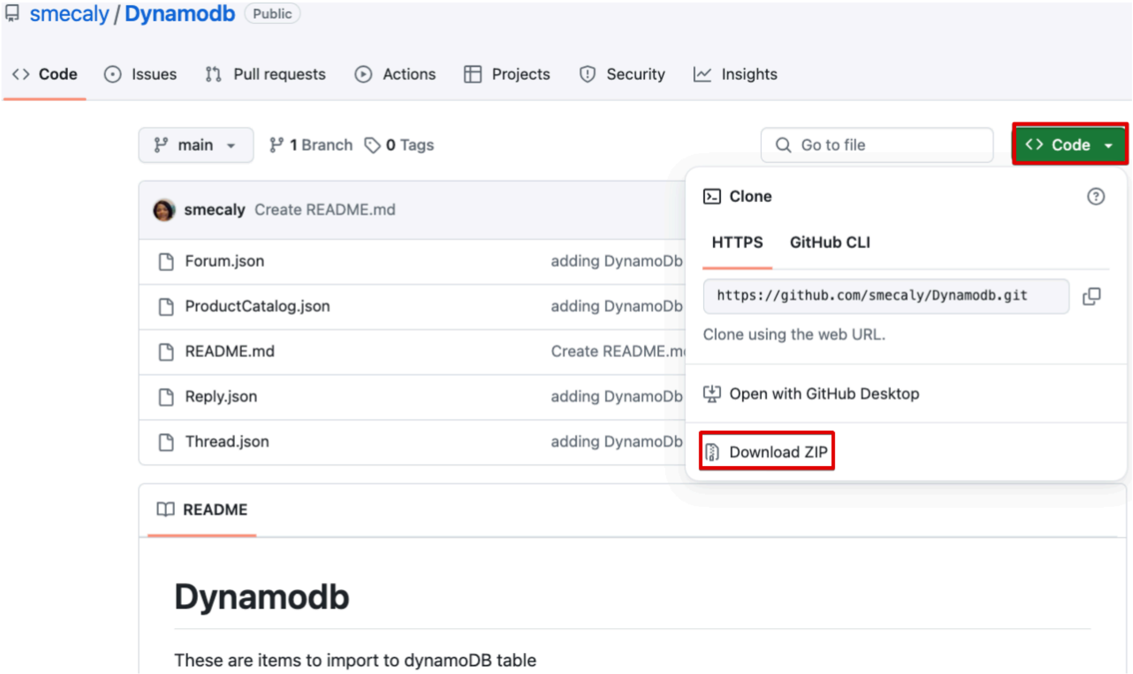
```
aws dynamodb describe-table --table-name ProductCatalog
aws dynamodb describe-table --table-name Forum
aws dynamodb describe-table --table-name Thread
aws dynamodb describe-table --table-name Reply
```



□	Name ▲	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mo...
□	Forum	✓ Active	Name (S)	-	0	⊖ Off	Provisioned (10)	Provisioned (5)
□	ProductCatalog	✓ Active	Id (N)	-	0	⊖ Off	Provisioned (10)	Provisioned (5)
□	Reply	✓ Active	Id (S)	ReplyDateTime (S)	0	⊖ Off	Provisioned (10)	Provisioned (5)
□	Thread	✓ Active	ForumName (S)	Subject (S)	0	⊖ Off	Provisioned (10)	Provisioned (5)

Step 4: Download Sample Data

1. **Download Sample Data**:
   - Go to https://github.com/smecaly/Dynamodb and download the ZIP file from the repository to your desktop.



smecaly Create README.md

Forum.json	adding DynamoDb
ProductCatalog.json	adding DynamoDb
README.md	Create README.m
Reply.json	adding DynamoDb
Thread.json	adding DynamoDb

https://github.com/smecaly/Dynamodb.git

Clone using the web URL.

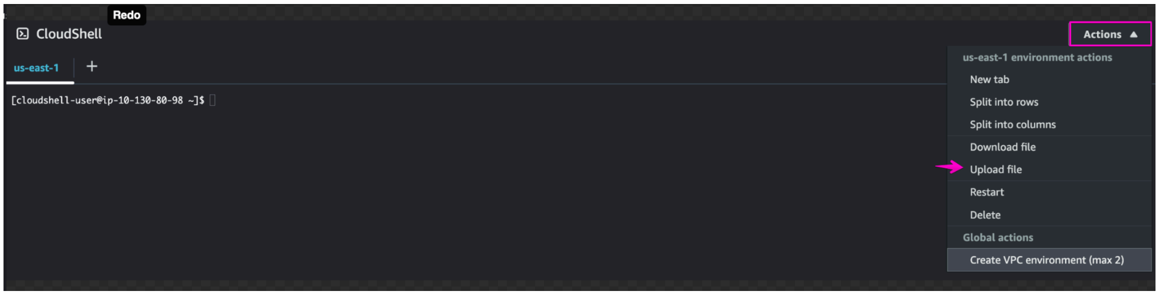
Dynamodb

These are items to import to dynamoDB table

   - Extract the ZIP file to access the sample data files.

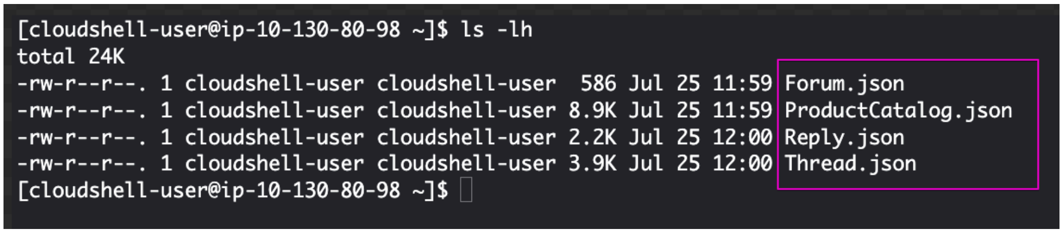
Step 5: Upload Sample Data to CloudShell

1. **Upload Data to CloudShell**:
   - In the AWS Management Console, open CloudShell.
   - Use the upload functionality in CloudShell to upload the sample data files from your local desktop.



```
[cloudshell-user@ip-10-130-80-98 ~]$
```

   - Example: Click on the upload button and select the data folder and the extracted files.

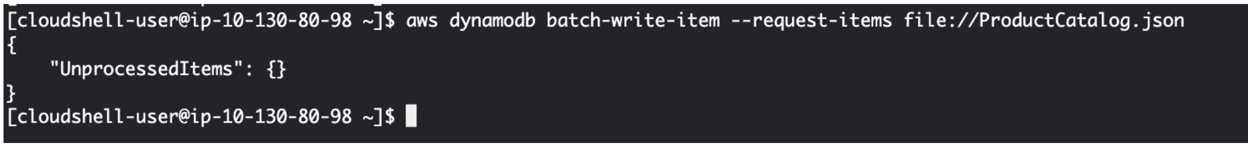


```
[cloudshell-user@ip-10-130-80-98 ~]$ ls -lh
total 24K
-rw-r--r--. 1 cloudshell-user cloudshell-user  586 Jul 25 11:59 Forum.json
-rw-r--r--. 1 cloudshell-user cloudshell-user 8.9K Jul 25 11:59 ProductCatalog.json
-rw-r--r--. 1 cloudshell-user cloudshell-user 2.2K Jul 25 12:00 Reply.json
-rw-r--r--. 1 cloudshell-user cloudshell-user 3.9K Jul 25 12:00 Thread.json
[cloudshell-user@ip-10-130-80-98 ~]$
```

Step 6: Load Sample Data into DynamoDB Tables

1. **Load Data into the ProductCatalog Table**:
   - Use the AWS CLI to load data into the ProductCatalog table:

     aws dynamodb batch-write-item --request-items file://ProductCatalog.json



```
[cloudshell-user@ip-10-130-80-98 ~]$ aws dynamodb batch-write-item --request-items file://ProductCatalog.json
{
    "UnprocessedItems": {}
}
[cloudshell-user@ip-10-130-80-98 ~]$
```

2. **Load Data into the Forum Table**:
   - Use the AWS CLI to load data into the Forum table:

     aws dynamodb batch-write-item --request-items file://Forum.json

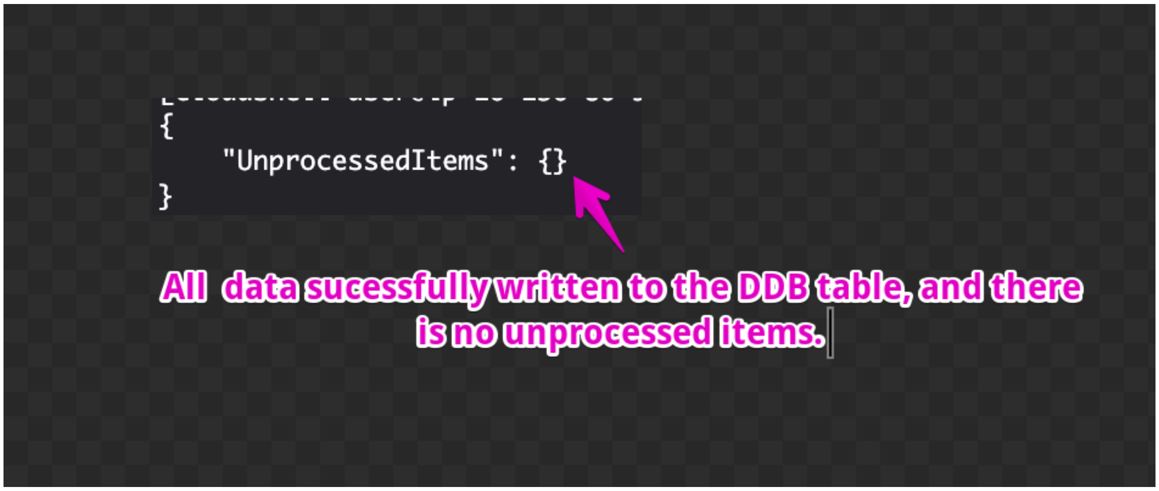
3. **Load Data into the Thread Table**:
   - Use the AWS CLI to load data into the Thread table:

     ```
     aws dynamodb batch-write-item --request-items file://Thread.json
     ```

4. **Load Data into the Reply Table**:
   - Use the AWS CLI to load data into the Reply table:

     ```
     aws dynamodb batch-write-item --request-items file://Reply.json
     ```

5. **Verify Data Load**:
   - After each data load, ensure there are no unprocessed items. You should see:



```
{
    "UnprocessedItems": {}
}
```

All data sucessfully written to the DDB table, and there is no unprocessed items.

Step 7: Enable Point-In-Time Recovery (PITR) Backups

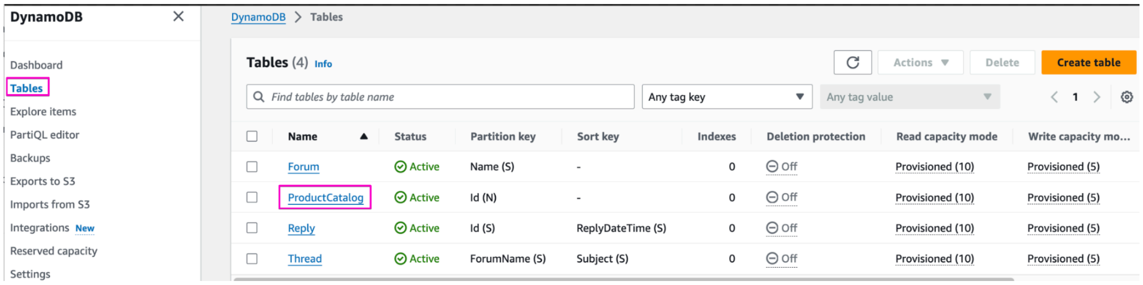
What is Point-In-Time Recovery (PITR)?

Point-In-Time Recovery (PITR) provides continuous backups of your DynamoDB table data. It allows you to restore your table to any second in time within the retention period, up to 35 days.

The goal is to ensure that your DynamoDB tables can be restored to any point in time within the last 35 days.

Steps to Enable PITR

1. **Enable PITR via the Console**:
   - Go to the DynamoDB Console.
   - Select the **ProductCatalog** table.

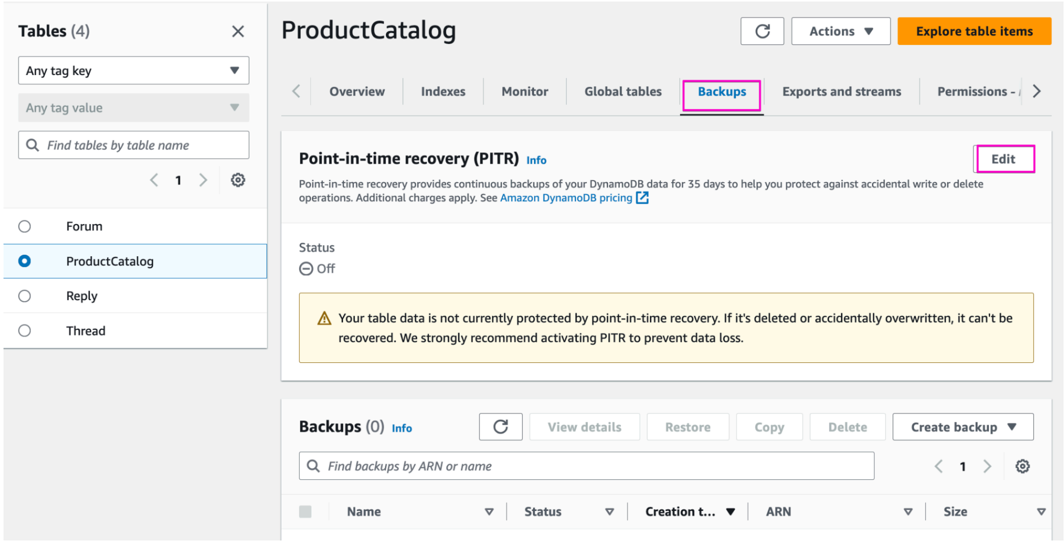


Tables (4) Info

Find tables by table name

□	Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mo...
□	Forum	Active	Name (S)	-	0	Off	Provisioned (10)	Provisioned (5)
□	ProductCatalog	Active	Id (N)	-	0	Off	Provisioned (10)	Provisioned (5)
□	Reply	Active	Id (S)	ReplyDateTime (S)	0	Off	Provisioned (10)	Provisioned (5)
□	Thread	Active	ForumName (S)	Subject (S)	0	Off	Provisioned (10)	Provisioned (5)

- Navigate to the **Backups** tab and click on **Edit**.



Point-in-time recovery (PITR) Info

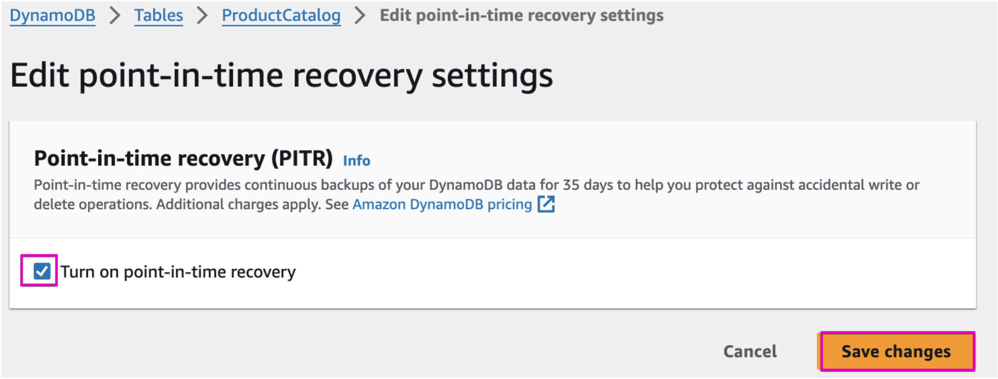
Point-in-time recovery provides continuous backups of your DynamoDB data for 35 days to help you protect against accidental write or delete operations. Additional charges apply. See Amazon DynamoDB pricing

Status
Off

Your table data is not currently protected by point-in-time recovery. If it's deleted or accidentally overwritten, it can't be recovered. We strongly recommend activating PITR to prevent data loss.

Backups (0) Info

- In the **Point-in-time recovery settings** section, **check the box** to *Turn on point-in-time recovery*, and **save changes**


DynamoDB > Tables > ProductCatalog > Edit point-in-time recovery settings
Cancel
Save changes

Edit point-in-time recovery settings

Point-in-time recovery (PITR) Info

Point-in-time recovery provides continuous backups of your DynamoDB data for 35 days to help you protect against accidental write or delete operations. Additional charges apply. See Amazon DynamoDB pricing

☑ Turn on point-in-time recovery

- Repeat these steps for the **Forum**, **Thread**, and **Reply** tables.

Step 8: Simulate On-Demand Backups

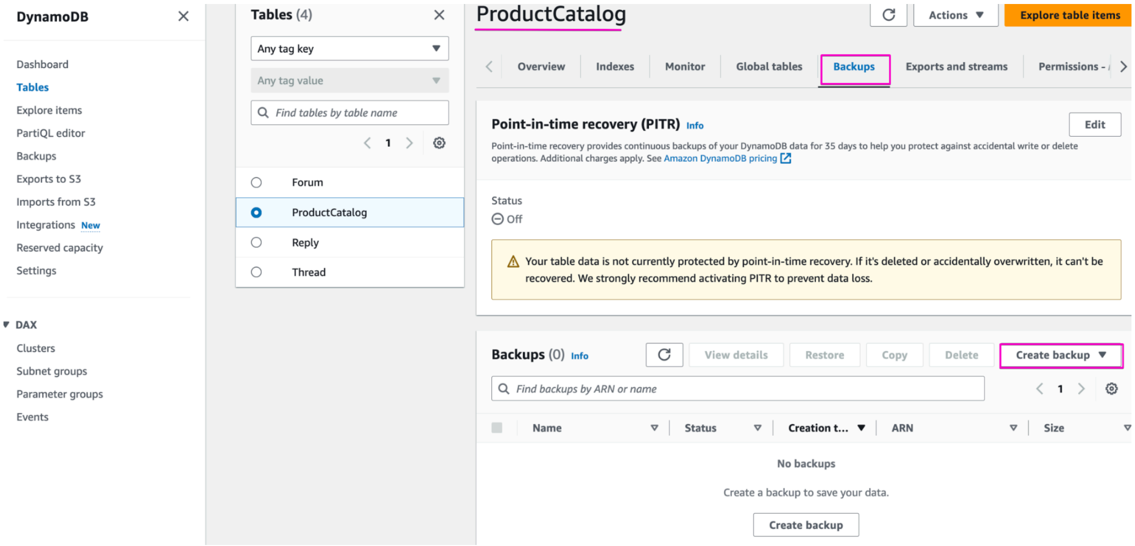
What are On-Demand Backups?

On-demand backups allow you to create full backups of your DynamoDB table data for long-term retention and archival purposes.

The goal is to create and verify on-demand backups of your DynamoDB tables.

Steps to Create an On-Demand Backup

1. **Create On-Demand Backup via the Console**:
   - Go to the [DynamoDB Console](#).
   - Select the **ProductCatalog** table.
   - Navigate to the **Backups** tab.
   - Click on **Create backup**.

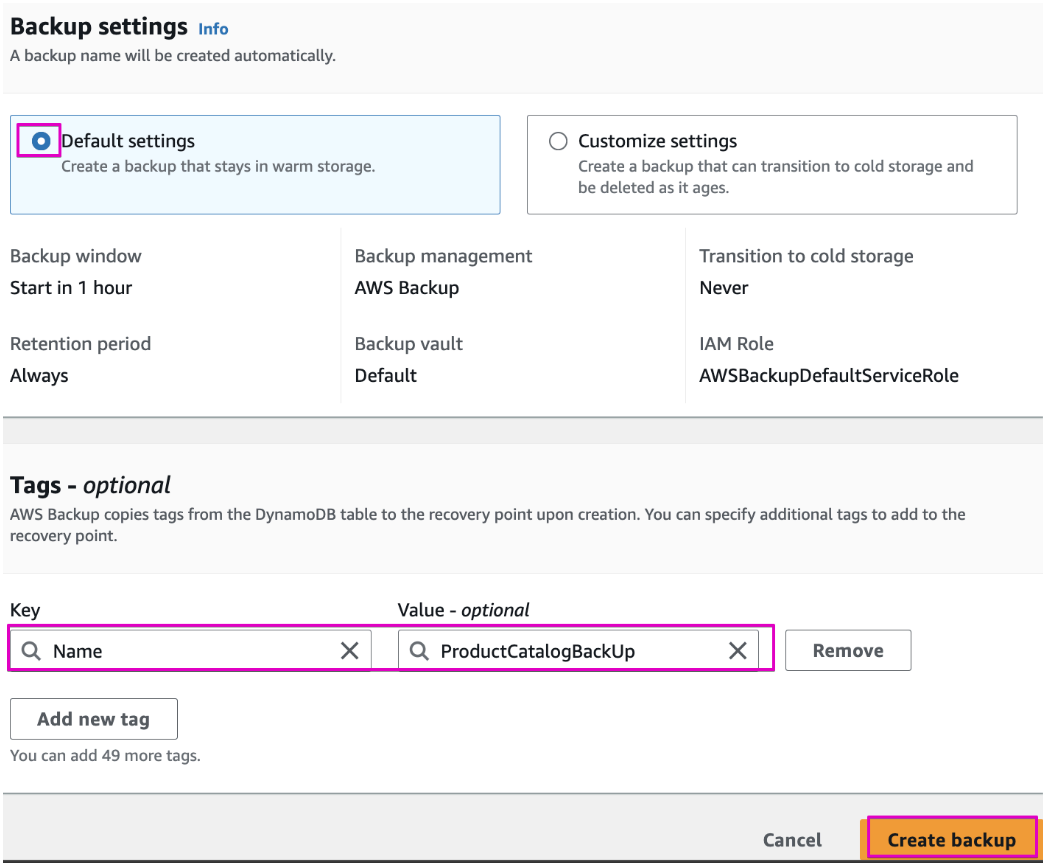


Point-in-time recovery provides continuous backups of your DynamoDB data for 35 days to help you protect against accidental write or delete operations. Additional charges apply. See Amazon DynamoDB pricing

Your table data is not currently protected by point-in-time recovery. If it's deleted or accidentally overwritten, it can't be recovered. We strongly recommend activating PITR to prevent data loss.

Create a backup to save your data.

   - select **Create on-demand backup**
   - Leave default settings to use AWS defaults or customize settings as per requirements.
   - Add a tag to the backup (e.g., **Name: ProductCatalogBackup**).
   - Click **Create**.



Backup settings Info

A backup name will be created automatically.

● Default settings
Create a backup that stays in warm storage.

○ Customize settings
Create a backup that can transition to cold storage and be deleted as it ages.

Backup window: Start in 1 hour

Backup management: AWS Backup

Transition to cold storage: Never

Retention period: Always

Backup vault: Default

IAM Role: AWSBackupDefaultServiceRole

Tags - *optional*

AWS Backup copies tags from the DynamoDB table to the recovery point upon creation. You can specify additional tags to add to the recovery point.

Key: Name

Value - *optional*: ProductCatalogBackUp

- Repeat these steps for the **Forum**, **Thread**, and **Reply** tables (e.g., ForumBackup, ThreadBackup, ReplyBackup).

2. **Verify the Backup**:
   - After the BackUp window, Ensure the backups appear in the list of backups with the status **Available**.