

# Lab 3 - Developing Flexible NoSQL Solutions with Amazon DynamoDB

Friday, December 25, 2020 10:43 AM

## Developing on AWS – Lab 3 – Developing Flexible NoSQL Solutions with Amazon DynamoDB

2 hours 30 minutes

Free

★★★★★ [Rate Lab](#)



training and  
certification

**.Net version**

© 2020 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Corrections, feedback, or other questions? Contact us at [AWS Training and](#)

Corrections, feedback, or other questions? Contact us at [AWS Training and Certification](#).

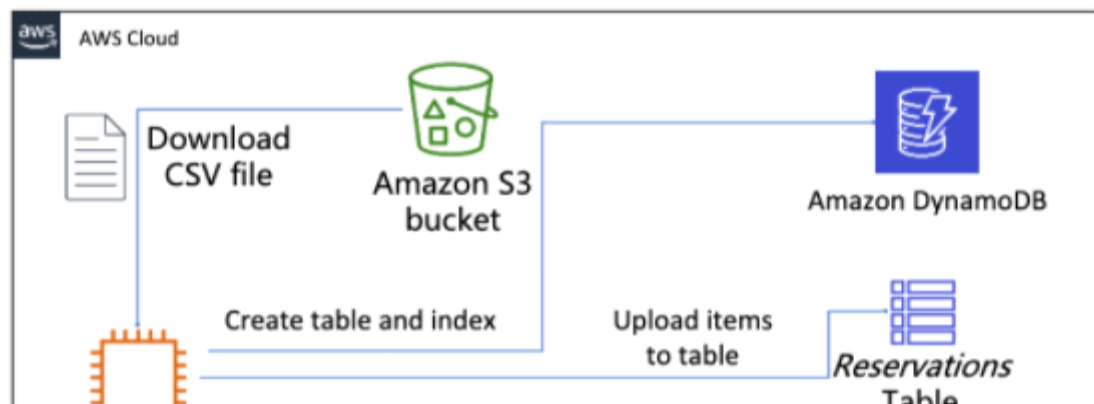
## Overview

In this lab, you will learn how to use the AWS SDK to create Amazon DynamoDB tables and indexes, add and update items, and query data.

You will use a DynamoDB table to store data for a travel agency that is booking reservations. An external system is already collecting the reservation information and exporting a CSV file that has the following information about each reservation:

- Customer ID
- City
- Date that the trip was booked

You will develop an application that retrieves the booking data CSV file from an S3 bucket and loads the data into a DynamoDB table. You will create and use any necessary secondary indexes to query the data and print a count of reservations for a particular city.





## Objectives

After completing this lab, you will be able to:

- Create a table and global secondary index.
- Add items to a table.
- Update items by adding additional attributes.
- Query data by using the table's global secondary index.

## Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi running Microsoft Windows or macOS.
- An Internet browser such as Chrome, Firefox, or IE9+. (previous versions of Internet Explorer are not supported)
- You will need either an SSH client, such as PuTTY, or a Microsoft Remote Desktop client to connect to your development EC2 instance.

### Note

You can use an iPad or tablet device to access these directions in the lab console.

## Duration

This lab will require around **60 minutes** to complete.

# Start Lab

1. At the top of your screen, launch your lab by choosing **Start Lab**

This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

**i** If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing **Open Console**

This opens an AWS Management Console sign-in page.

3. On the sign-in page, configure:

- **IAM user name:** `awsstudent`
- **Password:** Paste the value of **Password** from the left side of the lab page
- Choose **Sign In**

**⚠ Do not change the Region unless instructed.**

## Common Login Errors

**Error: You must first log out**

**Amazon Web Services Sign In**

You must first log out before logging into a different AWS account

## Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, [click here](#)

If you see the message, **You must first log out before logging into a different AWS account:**

- Choose **click here**
- Close your browser tab to return to your initial lab window
- Choose [Open Console](#) again

## Task 1: Connecting to Your Development Environment

4. To connect to your Dev instance, see the [Appendix: Connecting to Your Development Environment](#).

## Task 2: Developing Your Application

You will develop an application that adds, updates, and queries data in DynamoDB. You will use the DynamoDB [Document API in the AWS SDK for .NET](#) to develop the application.

### Lab Skeleton Code

## Lab Skeleton Code

Your lab skeleton code has been set up on the Amazon EC2 instance - **Windows Dev Instance**.

The base working directory is at the following location: `c:\temp\workdir`

### Note

The Lab skeleton code includes comments beginning with `// TODO` to help you quickly locate the sections of code you must modify.

## Solution Code

You can refer to the solution code in `Solution.cs`.

## Task 2.1: Create Table and Global Secondary Index

In this section, you will run code to create a DynamoDB table and global secondary index.

5. In Windows Explorer, navigate to `c:\temp\workdir\dynamoDBSharpLab`. Open the **dynamoDBSolution** solution file in Visual Studio.

### Note

If prompted with the **Welcome** window, select **Not now, maybe later** and then select **Start Visual Studio**.

6. **Build** the Solution in Visual Studio by pressing **Ctrl+Shift+B**. This will restore dependencies using the NuGet Package Manager and allow IntelliSense to operate correctly. You may see errors when you open Visual Studio, but once you have built the Solution, you shouldn't see any errors.
7. **Optional** - Open the `ReservationsTableCreator.cs` file. In the `CreateReservationsTableWithIndex` method, you will find an instance of



7. **Optional** - Open the `ReservationsTableCreator.cs` file. In the `CreateReservationsTableWithIndex` method, you will find an instance of `CreateTableRequest` class with the following settings:
- Table name: `ReservationsTableName` constant
  - Partition key (Hash key): `CustomerID`
  - Attribute definitions: Created previously
  - Provisioned throughput: Read and write capacity units of `5L` and `10L` respectively
  - Global secondary index: Created previously
8. Open the **Test Explorer** (*Test > Windows > Test Explorer*).
9. Expand `Lab3 > Lab3 > ReservationsTests`
10. Run the unit test by right-clicking on `TestTableCreation` and then select **Run Selected Tests**. The test should indicate that there were zero failures and that the table was created.
11. **Optional** - Choose **Services** and select **DynamoDB**.
- In the navigation pane, select **Tables** to check that the `Reservations` table has been created.

## Task 2.2: Upload Data to Table

In this section, you will develop code to retrieve the CSV-formatted reservations data file from an S3 bucket. For each reservation, you will add an item to the reservations table.

12. Open the `ReservationsDataUploader.cs` file in Visual Studio from within the `dynamoDBSolution` solution. You will need to update the code in this file to implement the features described in the following instructions, unless noted otherwise.
13. In the `Init` method, check that the `AmazonDvnamoDBClient` has been

13. In the `Init` method, check that the `AmazonDynamoDBClient` has been created.
14. In the `AddItemToTable` method, create an instance of a DynamoDB item with attribute values read from the reservations data file. Then add the item to the reservations data table.

### TODO in the code

```
// TODO 1: Replace the solution with your own code
```

For more information, see [Putting an Item](#).

15. Save the `ReservationsDataUploader.cs` file.
16. Build the solution, but do not run it, in Visual Studio and ensure no errors are displayed.
17. Within the **Test Explorer**, run the unit test `TestDataUploader` - the test should indicate that there were zero failures and that the data was uploaded to the table.

Select the **Output** link, located in the lower left corner of the screen, to see what data was uploaded.

In case of errors, correct the code and run this command again.

## Task 2.3: Run a Query

In this section, you will develop code to query the reservations table to determine the number of reservations reported in a particular city. You will retrieve a reference to the global secondary index that you created earlier. You will create a query specification and use it to invoke a query on the



retrieve a reference to the global secondary index that you created earlier. You will create a query specification and use it to invoke a query on the index. You will then iterate through the results returned by the query and calculate the total number of reservations for each city.

18. Open the `ReservationsStatistics.cs` file in Visual Studio from within the `dynamoDBSolution` solution. You will need to update the code in this file to implement the features described in the following instructions, unless noted otherwise.
19. Check that the `ReservationsTableName` variable contains a reference to the reservations table for use in the `QueryCityRelatedItems` method.
20. In the `QueryCityRelatedItems` method, using the `CityDateIndexName` constant as index name, create an instance of the `QueryRequest` class. The query should only return items where the value of the `City` attribute matches the `inputCity` parameter specified.

Then invoke the `Query` method of the `dynamoDBClient` object with the `QueryRequest` as argument.

For more information, see [Querying Using the AWS SDK for .NET Document API](#).

### TODO in the code

```
// TODO 2: Replace the solution with your own code
```

21. Save the `ReservationsStatistics.cs` file.
22. Build the solution, but don't run it, in Visual Studio and ensure no errors are displayed.
23. Within the **Test Explorer**, run the unit test `TestReservationsStatistics` - the test should indicate that there were zero failures.

Check the output to make sure that **178** reservations were reported for Paris.

the test should indicate that there were zero failures.

Check the output to make sure that **178 reservations** were reported for **Reno** city. In case of errors, correct the code and run this command again.

## Task 2.4: (Optional Challenge) Update Items

In this section, you will develop code to update items in the reservations table. You will retrieve a list of customer file names from an S3 bucket. You will then update the item for each customer to add a link to the corresponding customer report in the S3 bucket.

24. Open the `CustomerReportLinker.cs` file in Visual Studio from within the `dynamoDBSolution` solution. You will need to update the code in this file to implement the features described in the following instructions, unless noted otherwise.
25. In the `LinkCustomerReport` method, review the structure of the code. The method updates items related to customer `1`, `2`, and `3`.
26. In the `UpdateItem` method, update the item in the table.

### TODO in the code

```
// TODO 3: Replace the solution with your own code
```

27. Save the `CustomerReportLinker.cs` file.
28. Build the solution, but do not run it, in Visual Studio and ensure no errors are displayed.
29. Within the **Test Explorer**, run the unit test `TestCustomerReportLinker` - the test should indicate that there were zero failures.

In case of errors, correct the code and run this command again.

In case of errors, correct the code and run this command again.

## End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

30. Return to the AWS Management Console.

31. On the navigation bar, choose **awsstudent@<AccountNumber>**, and then choose **Sign Out**.

32. Choose  **End Lab**

33. Choose  **OK**

34. (Optional):

- Select the applicable number of stars ☆
- Type a comment
- Choose **Submit**
  - 1 star = Very dissatisfied
  - 2 stars = Dissatisfied
  - 3 stars = Neutral
  - 4 stars = Satisfied
  - 5 stars = Very satisfied

You may close the window if you don't want to provide feedback.  
Congratulations! You are done!

You may close the window if you don't want to provide feedback.  
Congratulations! You are done!

## Additional Resources

For more information about AWS Training and Certification, see <http://aws.amazon.com/training/>.

*Your feedback is welcome and appreciated.*

If you would like to share any feedback, suggestions, or corrections, please provide the details in our [AWS Training and Certification Contact Form](#).

## Appendix: Connecting to Your Development Environment

You can connect to your Dev instance by using one of the following methods:

- Use Apache Guacamole to connect to your Windows Dev instance
- Use Remote Desktop to connect to your Windows Dev instance

To connect to the **Windows EC2 instance** by using Guacamole (recommended), see the following directions:

[Connect to Your Windows Dev Instance by Using Apache Guacamole](#)

(recommended), see the following directions.

- [Connect to Your Windows Dev Instance by Using Apache Guacamole](#)

To connect to the **Windows EC2 instance** by using RDP, see the following directions:

- [Connect to Your Windows Dev Instance from a Windows Machine](#)
- [Connect to Your Windows Dev Instance from a macOS Machine](#)

## Connect to Your Windows Dev Instance by Using Apache Guacamole

35. In the **Connection Details** section in the lab console, go to the bottom for the Guacamole information. Copy the **GuacamoleLink** and paste it into a browser.
36. Go back to the lab console and copy the **WindowsPassword** to the clipboard.
37. Go to the Apache Guacamole sign in in the browser. Sign in by using the following steps:
  - For **Username**, enter: `student`
  - For **Password**, paste the **WindowsPassword** from the clipboard.
  - Select **Log In**.

Your connection to your remote instance should start momentarily. Once you open a connection, you will see an image of the Dev instance desktop. You can interact with this image just as you would your normal desktop, or any remote desktop client.

remote desktop client.

You are now connected to your Windows Dev instance in the browser via Guacamole.

**Tip** Web browsers don't provide access to clipboard data, which means synchronization between your local clipboard and the remote clipboard is impossible. To copy and paste when using Guacamole, you must use the Clipboard editor. To open the Clipboard editor, press **Ctrl -> Alt -> Shift**.

Copy your text and paste it to the Clipboard editor. This will set the clipboard of your Dev instance to what you just pasted. You can also edit the text that you place in the Clipboard editor before pasting into your remote desktop. To close the Clipboard editor, press **Ctrl -> Alt -> Shift**.

To continue this lab, move on to [Task 2: Developing Your Application](#).

## Connect to Your Windows Dev Instance from a Windows Machine

In this task, you will connect to a Windows EC2 instance from your Windows machine.

### Note

Perform the steps in this task only if you are connecting to **Windows Dev Instance** from a Windows machine.

38. In the lab console, go to the **Connection Details** section and copy the **WindowsInstanceIP** to the clipboard.
39. Open the Remote Desktop Connection application on your computer.



39. Open the Remote Desktop Connection application on your computer.

- On Windows 7, select the **Start** icon, and in the **Search programs and files** textbox, enter: `Remote Desktop Connection`. Select the application when it appears in the **Programs** list.
- On Windows 8, activate the Charms menu by moving the cursor into the lower right corner of the screen, and select the **Search** icon. Enter: `Remote Desktop Connection`. Select the application when it appears in the **Programs** list.
- On Windows 10, select the **Start** icon, and then, select the **Search** icon. Enter: `Remote Desktop Connection`. Select the application when it appears in the **Programs** list.

40. In Remote Desktop Connection, for **Computer**, paste the IP of your Windows instance that you copied.

41. Select **Connect**.

42. Remote Desktop Connection will prompt you with a login dialog asking for your username and password. By default, the application will use your current Windows username and domain. To change this, select **Use another account**.

#### **Note**

On Windows 10, select **More Choices** before selecting **Use a different account**.

43. Go back to the lab console and copy the **WindowsPassword** to the clipboard.

44. For your login credentials, use the following values:

- For **User name**, enter: `\Administrator`
- For *\*Password\**, paste the password from the clipboard.

#### **Note**

The `\` in the user name is important. as it tells Remote Desktop Connection

### **Note**

The `\` in the user name is important, as it tells Remote Desktop Connection that you are logging in as the local Administrator, and not as a domain user.

45. To connect to your instance, select **OK**. If you receive a prompt that the certificate used to verify the connection was not a known, trusted root certificate, select **Yes**.

### **Result**

Your connection to your remote instance should start momentarily. When lab instructions in subsequent sections require a command window, open or use a Powershell window.

To continue this lab, move on to [Task 2: Developing Your Application](#).

## **Connect to Your Windows Dev Instance from a macOS Machine**

In this section, you will connect to a Windows EC2 instance from your macOS machine.

46. In the lab console, go to the **Connection Details** section and copy the **WindowsInstanceIP** to the clipboard.
47. Install Microsoft Remote Desktop if it is not already installed.
  - From the Dock, launch **App store**.
  - Search for the following string: `Microsoft Remote Desktop`
  - Select **Install**.

• Search for the following string: `Microsoft Remote Desktop`

- Select **Install**.

48. To open **Microsoft Remote Desktop**, on the Dock, select **Launchpad**. Then, select **Microsoft Remote Desktop**.

49. To create a new connection, select **New**.

50. Use the following values:

- For **Connection name**, enter: `Windows Dev Instance`
- For *\*PC Name\**, paste in the IP address of your Windows Server instance that you copied to the clipboard.
- For *\*User name\**, enter: `\Administrator`

51. Go back to the lab console and copy the **WindowsPassword** to the clipboard.

52. Go back to your Microsoft Remote Desktop connection window and enter the following value:

- For *\*Password\**, paste in the password that you copied to the clipboard.

### **Note**

The `\` in the user name is important, as it tells Remote Desktop Connection that you are logging in as the local Administrator, and not as a domain user.

53. Close the *Edit Remote Desktops* window by selecting the button on the top left corner.

54. In the *Microsoft Remote Desktop* window, select the connection titled **Windows Dev Instance** and select **Start**.

55. In the *Verify Certificate* dialog, select **Continue** to complete the connection.

## **Result**

Your connection to your remote instance should start momentarily. When lab instructions in subsequent sections require a command window, open or use

Your connection to your remote instance should start momentarily. When lab instructions in subsequent sections require a command window, open or use a Powershell window.

To continue this lab, move on to [Task 2: Developing Your Application](#).

## Appendix: Configuring AWS SDK for .NET Credentials

The following resources are excerpts from the [AWS SDK for .NET Developer Guide](#).

### AWS Credentials

You can manage credentials for your AWS SDK for .NET application in the following ways:

- [Using the SDK Store](#)
- [Using a Credentials File](#) ( `C:\Users\<username>\.aws\credentials` )

For more information, see: [AWS Access Keys Best Practices](#)

To continue this lab, move on to [Task 2: Developing Your Application](#).

