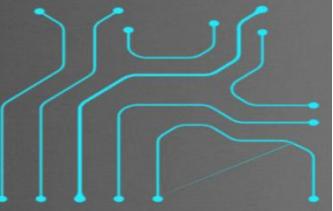


# Demonstration For IAM and Its Best Practices



- 1.Create Dynamodb table
- 2.Create IAM User and Attached policy
- 3.Verify DynamodbTable
- 4.Best Practices for IAM



[linkedin.com/in/vyankateshwar-taikar/](https://linkedin.com/in/vyankateshwar-taikar/)

## Table of contents

### ★ Create Dynamodb Table

### ★ Create IAM User and Attach Policy

### ★ IAM Best Practices

## Practical Implementation of IAM(Step by Step Guide)

### ★ Create Dynamodb Table

#### 1. Initially open the DynamoDB Dashboard.

Go to <https://aws.amazon.com/> to access the AWS Management Console.

Type "DynamoDB" into the "Find Services" search bar and click on it.

The screenshot shows the AWS Management Console interface. In the top navigation bar, 'Services' is selected. A search bar contains the text 'Dynamodb'. Below the search bar, the results for 'Dynamodb' are displayed under the 'Services' category. The 'DynamoDB' service is highlighted with a red box. To the left, there's a sidebar with links like 'Launch', 'Name an instance', and 'Marketplace'. On the right, there's a large empty area for displaying service-specific content.

#### 2. Click the "Create table" button in the DynamoDB console.

The screenshot shows the Amazon DynamoDB console. At the top, there are links for S3, Amazon Q, and DynamoDB, along with a feedback survey. The main heading is 'Amazon DynamoDB' with the subtext 'A fast and flexible NoSQL database service for any scale'. Below this, a 'Get started' section has a button labeled 'Create table'. At the bottom, there's a note about DynamoDB being a fully managed, key-value, and document database.

### 3. Give Table Specifics Name of Table & Primary key (Partition key) :

Enter the attribute name and choose the data type to specify the primary key for your table. The primary key for this could be either a single attribute (the partition key) or a combination of attributes (the sort key plus the partition key).

(Optional) If necessary, configure other parameters like encryption, auto-scaling, and provided throughput. Select "Create" from the menu.

DynamoDB > Tables > Create table

### Create table

**Table details** [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

**Table name**  
This will be used to identify your table.  
 Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.)

**Partition key**  
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.  
 [String](#) ▾  
1 to 255 characters and case sensitive.

**Sort key - optional**  
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.  
 [String](#) ▾  
1 to 255 characters and case sensitive.

**Tags**  
Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

[Add new tag](#)  
You can add 50 more tags.

[Cancel](#) [Create table](#)

4. Wait for Table Creation DynamoDB will start creating the table. The status will initially be "Creating." Wait for the status to change to "Active." This indicates that the table is ready for use.

**Share your feedback on Amazon DynamoDB** Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing. [Share feedback](#) [X](#)

**Creating the Employee table. It will be available for use shortly.** [X](#)

DynamoDB > Tables

**Tables (1)** [Info](#)

<input type="checkbox"/>	Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode
<input type="checkbox"/>	Employee	<a href="#">Creating</a>	EmpID (S)	-	0	<a href="#">Off</a>	Provisioned (5)	Provisioned (5)

## 5. Table Created :

Bravo! A DynamoDB table has been successfully created. At this stage, you are able to start adding items to the table and customizing other parameters as necessary.

The screenshot shows the AWS DynamoDB 'Tables' page. A green success message at the top states: 'The Employee table was created successfully.' Below it, the 'Tables (1) Info' section displays the 'Employee' table details. The table has one item: EmpID (S) with a value of '1'. Other columns include Name (Employee), Status (Active), Partition key (EmpID S), Sort key (-), Indexes (0), Deletion protection (Off), Read capacity mode (Provisioned (5)), and Write capacity mode (Provisioned (5)).

👉 To know more about DynamoDB [Click Here](#) pr <https://aws.amazon.com/dynamodb/>

### 1. ADD Data in Table : Goto Dynamodb > Explore Items > Employee

The screenshot shows the 'Explore Items' page for the 'Employee' table. On the left, the 'Tables (1)' sidebar shows the 'Employee' table selected. On the right, the main panel is titled 'Employee' with 'Autopreview' turned on. It features a 'Scan or query items' section with 'Scan' selected. Below it, 'Select a table or index' is set to 'Table - Employee' and 'Select attribute projection' is set to 'All attributes'. At the bottom of this section are 'Run' and 'Reset' buttons. A green message box indicates 'Completed. Read capacity units consumed: 0.5'. In the bottom right corner, there is a 'Create item' button with a red box around it, and below it, a link 'Click Here' with a smiley face icon.

- After create item click on JSON View and insert the below data. or directly add.

The screenshot shows the 'Create item' screen in the AWS DynamoDB console. In the 'Attributes' section, there is a table with one row. The first column is 'Attribute name' (EmpID - Partition key) and the second column is 'Value' (1111). A red box highlights the 'Value' input field. In the top right corner, there are two buttons: 'Form' (blue) and 'JSON view' (white). Below the table, there is a 'Scan or query items' section with a 'Scan' button selected. Under 'Scan', it says 'Select a table or index' (Table - Employee) and 'Select attribute projection' (All attributes). A green message at the bottom indicates 'Completed. Read capacity units consumed: 0.5'. At the bottom, there is a table titled 'Items returned (1)' with one row: 'EmpID (String)' with value '1111'. A red box highlights this row. On the far right, there are 'Actions' and 'Create item' buttons, with 'Create item' also highlighted by a red box.

Now create IAM user, Give permission as a **AmazonDynamoDBReadOnlyAccess** and try to change in Dynamodb table. so it should have to give gives error as **permission denied**. let's check it out

## ★ Create IAM User and Attach Policy

- Go the IAM :

navigate to the Amazon Management Console. Type "IAM" into the "Find Services" search bar and click on it.

The screenshot shows the 'Services' page in the AWS IAM console. The search bar at the top contains 'IAM'. On the left, there is a sidebar with links: 'Services (11)', 'Features (21)', 'Resources New', 'Documentation (49,027)', 'Knowledge Articles (549)', and 'Marketplace (722)'. The main area is titled 'Services' and shows a card for 'IAM' with the subtext 'Manage access to AWS resources'. Below this, there is a 'Top features' section with links for 'Groups', 'Users', 'Roles', 'Policies', and 'Access Analyzer'. A 'See all 11 results ▾' link is in the top right corner.

- Keep going to "Users": Select "Users" from the left navigation pane of the IAM console. Click "Add user".

The screenshot shows the 'Identity and Access Management (IAM)' dashboard. At the top, it says 'Identity and Access Management (IAM)'. Below that is a search bar with 'Search IAM'. The main navigation menu on the left includes 'Dashboard' and 'Access management'. Under 'Access management', there are links for 'User groups' and 'Users'. The 'Users' link is underlined, indicating it is selected.

- Click "Create user" from the menu. Enter the user's data.
- Give the **IAM** user a **username**. Select the kind of access. Choose "Programmatic access" (for AWS CLI, SDK, etc.) for this scenario. If you would like to **grant console access**, you can select "AWS Management Console access" as an option.

**Note :** For Programmatic access you can generate access keys after you generate.

## Specify user details

### User details

User name

DevUser1

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a **best practice** [to manage their access in IAM Identity Center](#).



Are you providing console access to a person?

User type

Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

### Console password

Autogenerated password

You can view the password after you create the user.

Custom password

Enter a custom password for the user.

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & \* ( ) \_ + - (hyphen) = [ ] { } | '

Show password

Users must create a new password at next sign-in - Recommended

Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel

Next

4. \*\*Configure Permissions:\*\* Go to "Set permissions" and select "Attach existing policies directly." Look for the "**AmazonDynamoDBReadOnlyAccess**" policy and attach it. DynamoDB can be accessed read-only with this policy.

The screenshot shows the "Set permissions" step in the AWS IAM console. On the left, a sidebar lists "Step 3: Review and create" and "Step 4: Retrieve password". The main area is titled "Permissions options" and contains three choices: "Add user to group", "Copy permissions", and "Attach policies directly" (which is selected and highlighted with a red box). Below this is a search bar with "AmazonDynamoDBReadOnlyAccess" typed in, and a table showing the results. The first result, "AmazonDynamoDBReadOnlyAccess", is selected and highlighted with a red box. At the bottom right of the main area are "Cancel", "Previous", and "Next" buttons, with "Next" also highlighted with a red box.

5. \*\*Review:\*\* Check permission settings and user information. If you would like to add tags, click the "Add tags" button (optional).

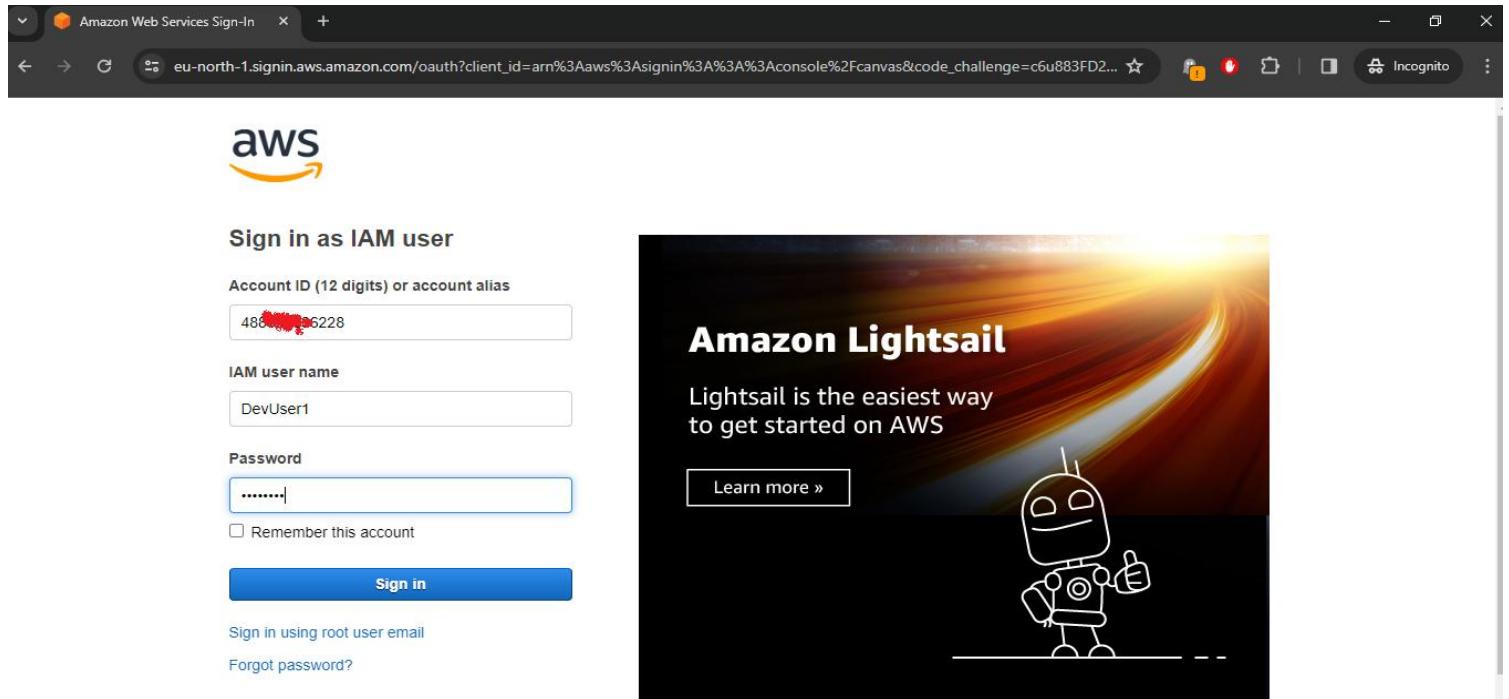
**Create User:** Select "Create user" from the menu.

The screenshot shows the "Create user" step in the AWS IAM console. On the left, a sidebar lists "Step 1: Specify user details", "Step 2: Set permissions" (which is selected and highlighted with a red box), "Step 3: Review and create", and "Step 4: Retrieve password". The main area is divided into several sections: "User details" (User name: DevUser1, Console password type: Autogenerated, Require password reset: Yes); "Permissions summary" (listing "AmazonDynamoDBReadOnlyAccess" and "IAMUserChangePassword" policies); "Tags - optional" (No tags associated with the resource, with an "Add new tag" button); and "Create user" at the bottom right, which is highlighted with a red box. Navigation buttons "Cancel", "Previous", and "Create user" are at the bottom right.

6. \*\*Retrieve password :\*\*The user's password can be viewed and downloaded below, and you can also send the user login credentials to access the **AWS Management Console**. You can only view and download this password at this time.

**Note: Download .csv file and keep safe in one place.**

**(In that file:** Console sign-in URL, User name, Password using you can login & **check that you are able to change the data in Dynamodb table.** let's do it.



After it change your password & go to Dynamodb table.(you should be in a same region where you created Dynamodb)

A screenshot of the AWS DynamoDB console. The top navigation bar shows 'Services' selected. The main area is titled 'Tables (1) Info' and shows a single table named 'Employee'. The table details are: Name: Employee, Status: Active, Partition key: EmpID (\$), Sort key: -, Indexes: 0, Deletion protection: Off, Read capacity mode: Provisioned (1), Write capacity mode: Provisioned (1). A red box highlights the 'Employee' table name. The top right corner of the main window has a red box around the 'N. Virginia' dropdown menu, which is currently set to 'DevUser1 @ 4886-5568-622'. A blue banner at the top says 'Share your feedback on Amazon DynamoDB' with a 'Share feedback' button.

Go to Employee > Explore table item & try to delete it .

The screenshot shows the AWS DynamoDB console interface. At the top, there is a feedback banner: "Share your feedback on Amazon DynamoDB" with a link to "Share feedback". Below it, an error message is displayed in a red box: "Your delete item request encountered issues. User: arn:aws:iam::488655686228:user/DevUser1 is not authorized to perform: dynamodb>DeleteItem on resource: arn:aws:dynamodb:us-east-1:488655686228:table/Employee because no identity-based policy allows the dynamodb>DeleteItem action." The main area shows the "Employee" table details. On the left, a sidebar titled "Tables (1)" lists the "Employee" table. The main panel has tabs for "Scan or query items" (selected), "Query", and "Filters". It includes dropdowns for "Select a table or index" (set to "Table - Employee") and "Select attribute projection" (set to "All attributes"). Below these are "Run" and "Reset" buttons. A success message at the bottom states: "Completed. Read capacity units consumed: 0.5". The "Items returned (1/1)" section shows one item with attribute "EmpID (String)" having value "1111".

As you try to delete the Item, you got error . means you have only Read only access .Hope you understand the IAM where you can restrict the user by attaching the policy.

## ⭐ IAM Best Practices



Let's discuss a few best practices that can help you secure your AWS resources before we end up:

1. **Use of the root account should only be done when absolutely required:** The root account should not be used for routine administrative tasks. It's excellent practice to create IAM users with least privilege access because the root account user has default access to all resources for all Amazon services.
2. **Keep to the least privilege concept and regularly verify all IAM permissions:** The security principle of least privilege, which states that it is preferable to deny access to a user to a resource if they do not need to interact with it, must be adhered to. Steer clear of using policy statements that provide access to all actions, all principals, or all resources because IAM permissions allow for very fine-grained access controls. Additionally, to make sure that a certain user is using all of the permissions assigned to them, make regular use of the IAM Access Advisor.
3. **MFA(Multi-Factor Authentication)-:**In addition, unless required, avoid creating access keys for the root account. Lastly, make sure hardware-based MFA is configured for root account access and set up monitoring to identify and notify on root account activity.

**For enhanced protection when interacting with the AWS API, turn on multi-factor authentication (MFA).**

4. **Use temporary login credentials:** Don't provide your login details to anyone. For anyone who needs access, it's best to create separate users, and it's even better to utilize temporary credentials. One excellent solution to this is to use dynamically created credentials that expire after a specified amount of time. For comprehensive details on this, see our hands-on tutorial on Securing Multi-Account Access on AWS.
5. **Require strong passwords:** You may regulate strong passwords by setting up an account password policy that limits the use of special characters to alphanumeric characters, and rotations passwords, and prevents the use of outdated passwords.
6. **Verify that the least privilege concept is used in both directions:** The access policy of many AWS resources, including S3 buckets, can be directly attached. Never make the mistake of assuming that just because an IAM role offers highly specific permissions and access is tightly controlled in one direction, you should relax your restrictions in the other direction (**for instance, when an S3 bucket access policy allows read access to all groups in your account**). To get the best results, make optimal use of both sides of the least privilege principle.

I hope you enjoy the blog post!