Table of contents

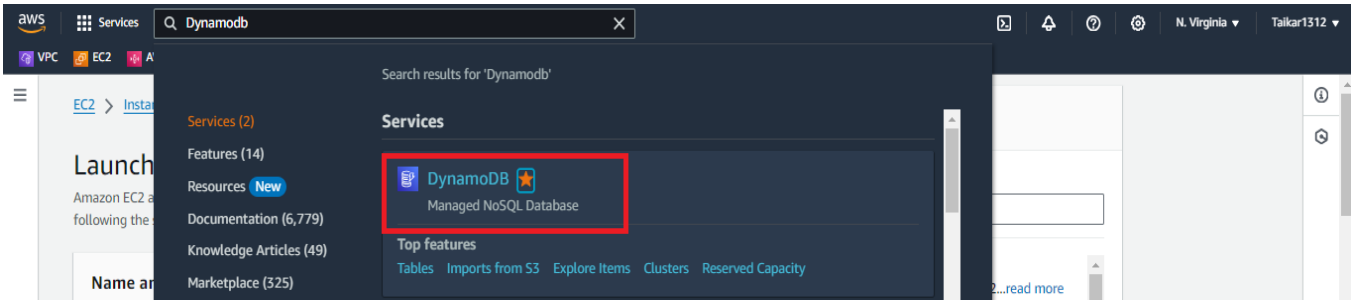
Practical Implementation of IAM(Step by Step Guide)

⭐ Create Dynamodb Table

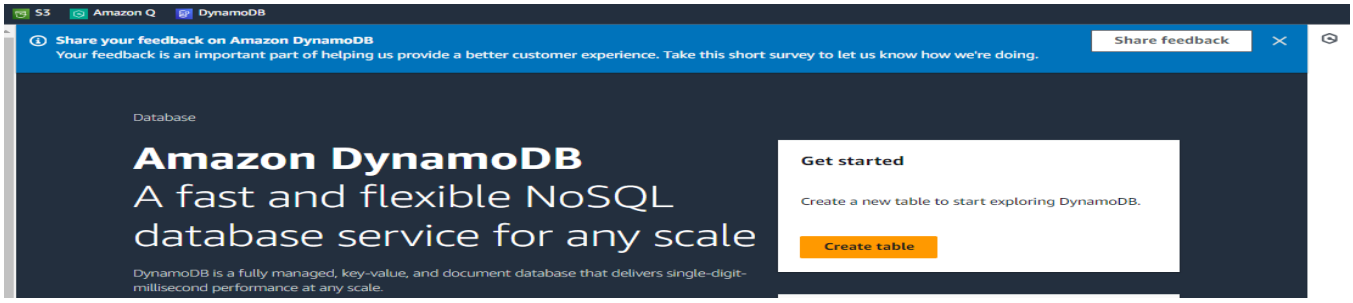
1. **Initially open the DynamoDB Dashboard.**

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

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



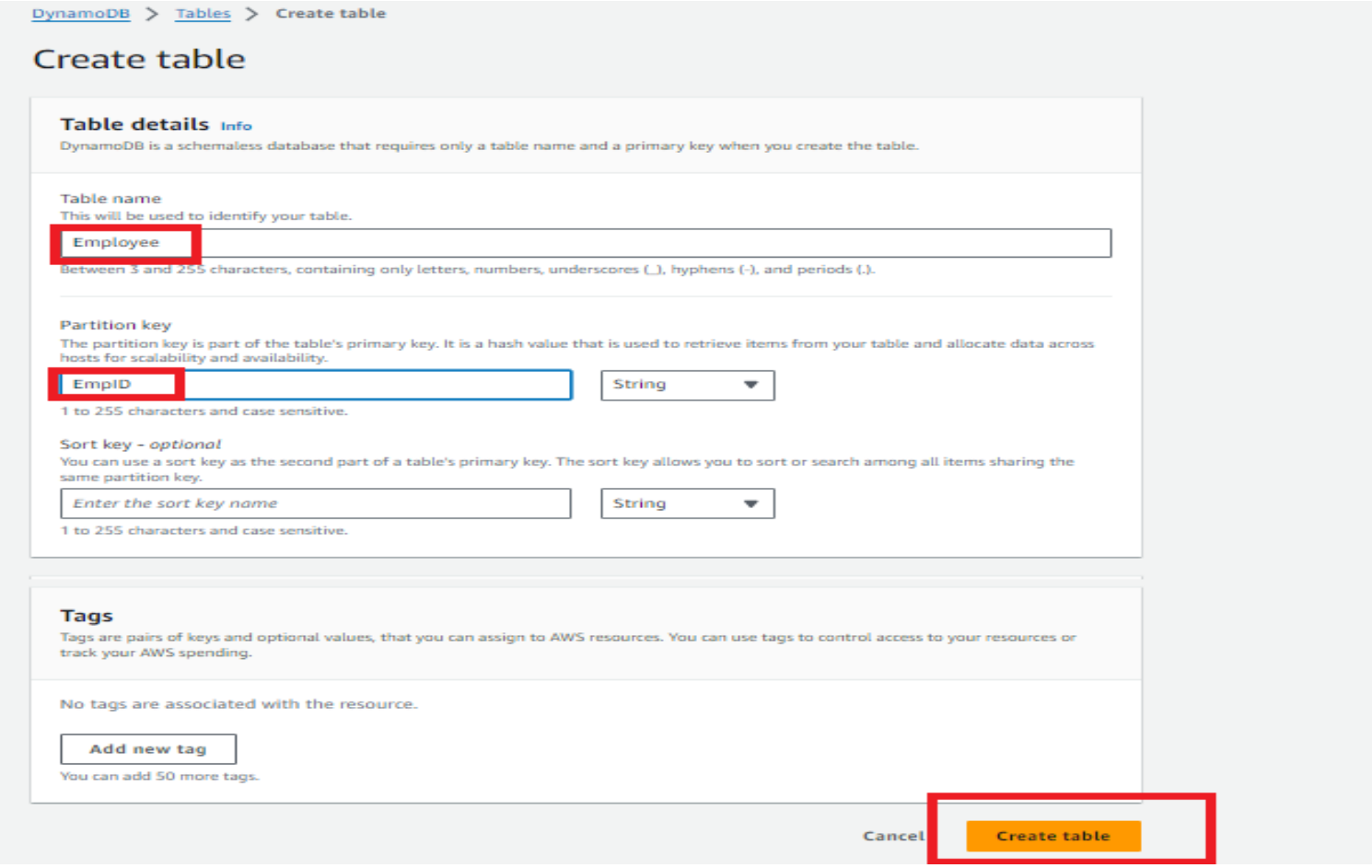
2. Click the "**Create table**" button in the DynamoDB console.



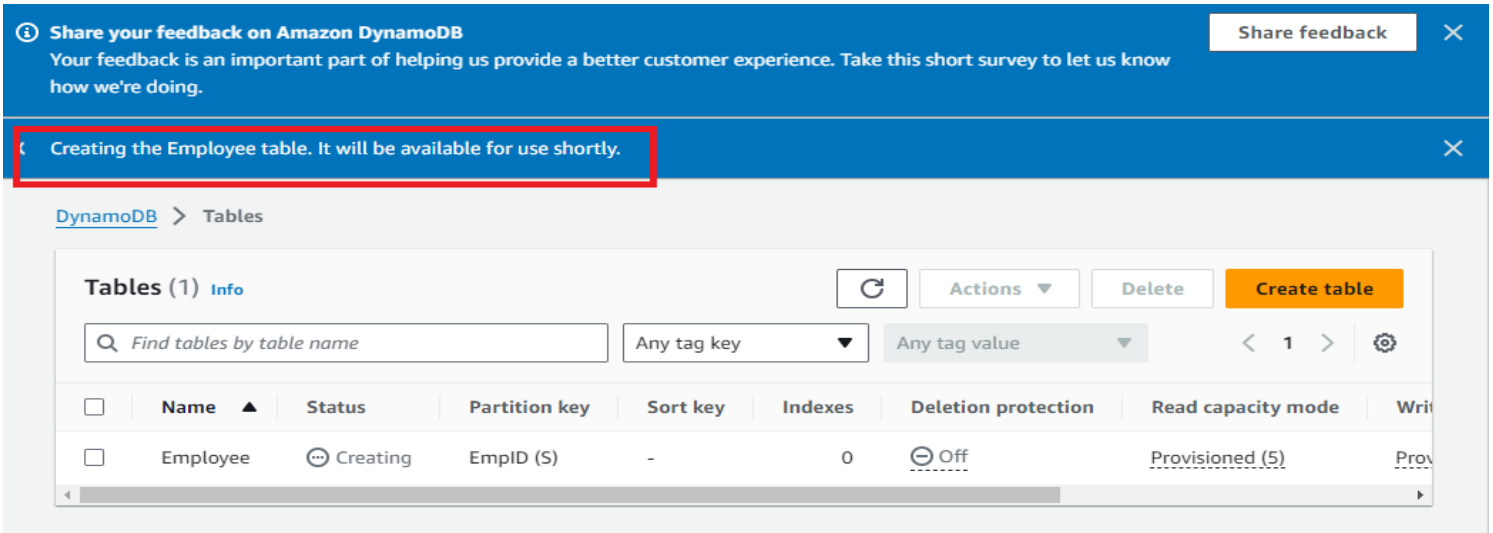
3. **Give Table Specifics Name of Table & Primary key (Partition key) :**

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.

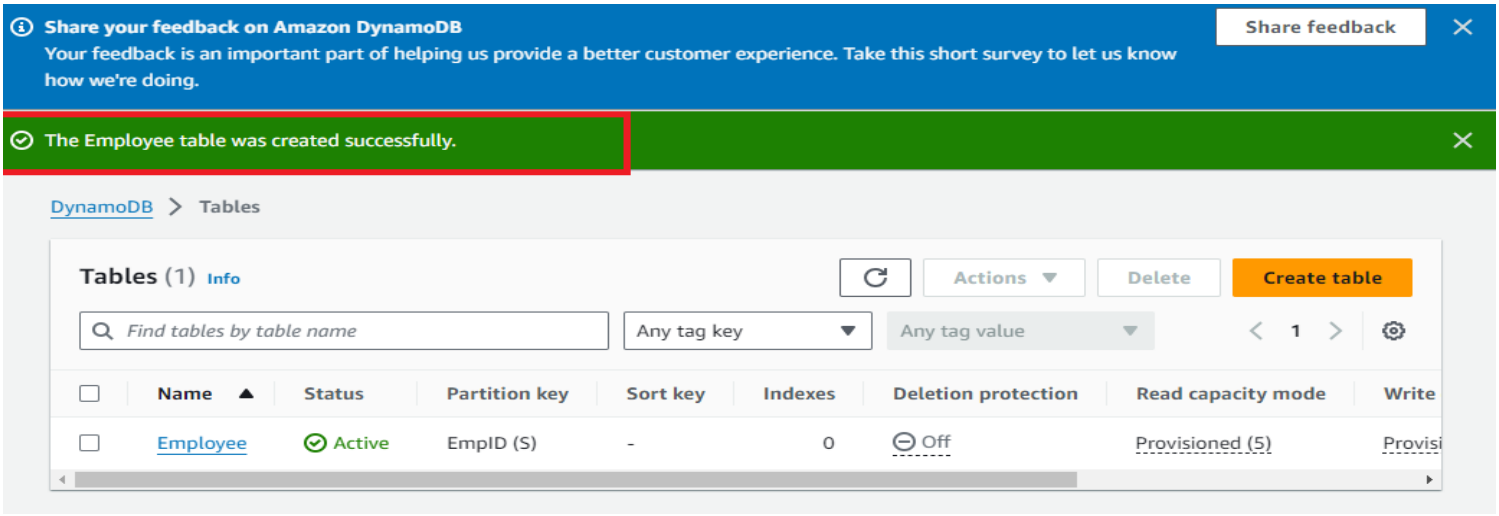


4. Wait for Table Creation DynamoDB will start creating the table. The status will initially be "Creating." Wait for the status to change to "Active." This indicates that the table is ready for use.



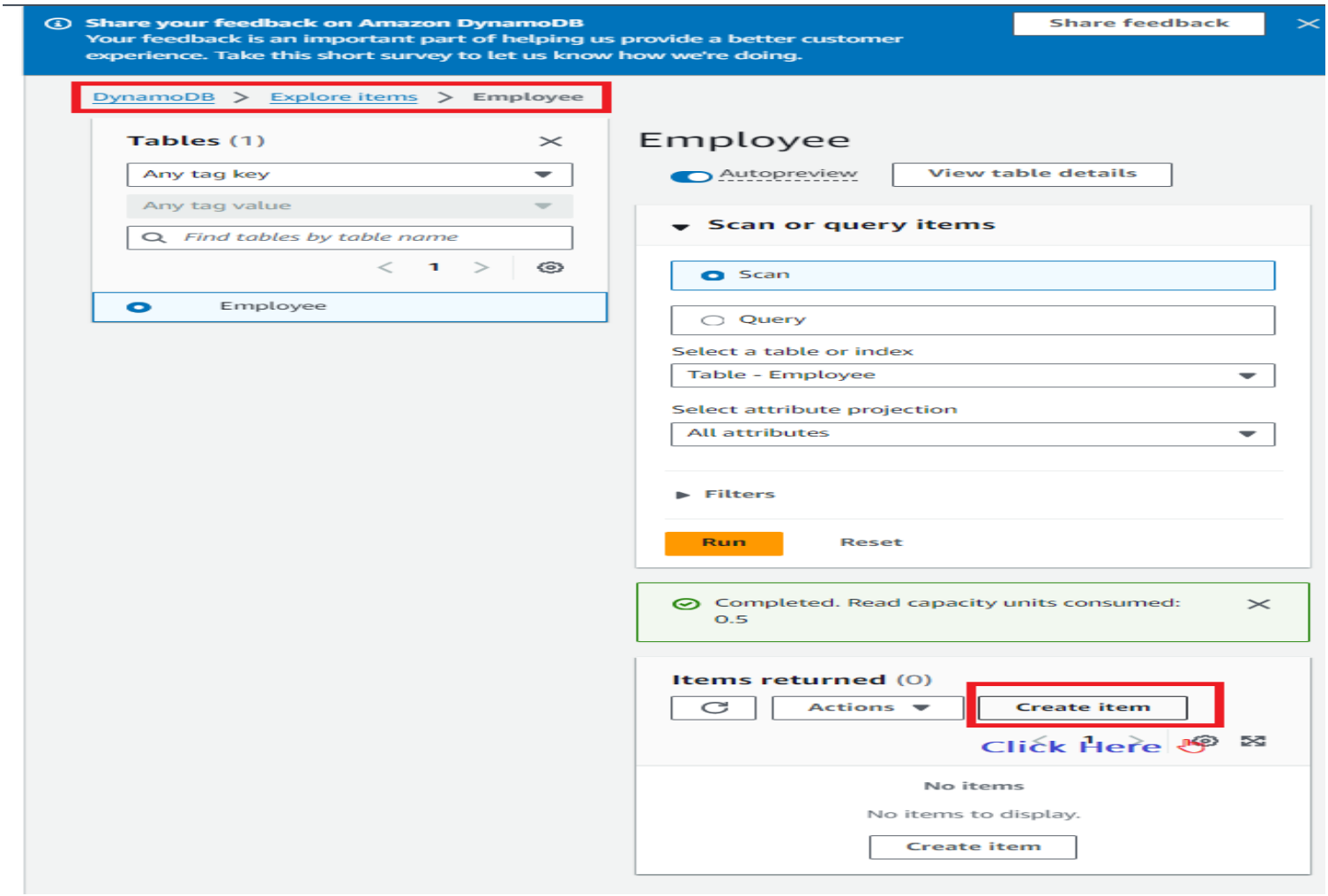
5. **Table Created :**

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.

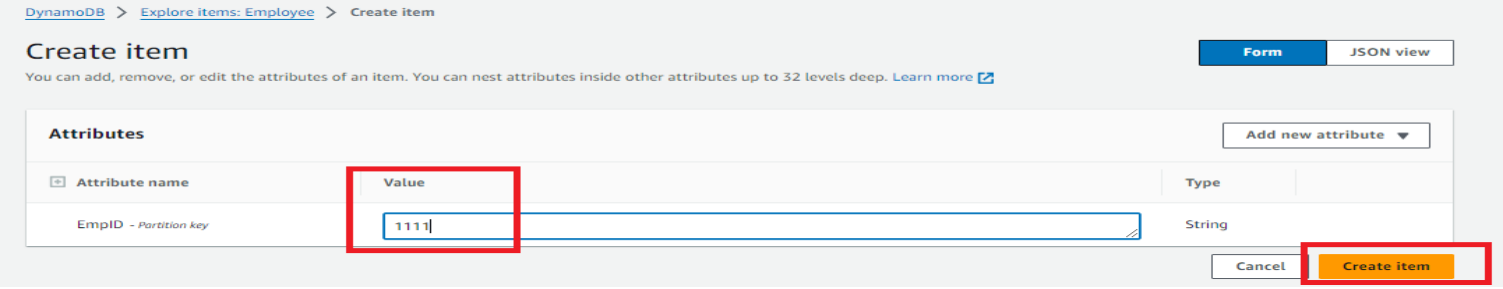


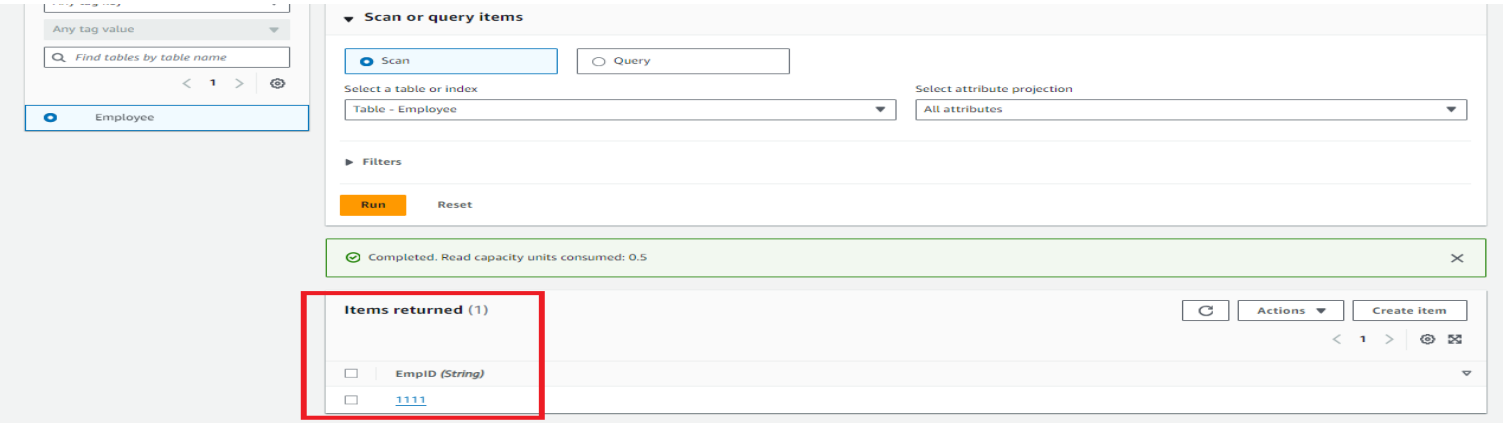
👉To know more about DynamoDB Click Here pr https://aws.amazon.com/dynamodb/

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



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



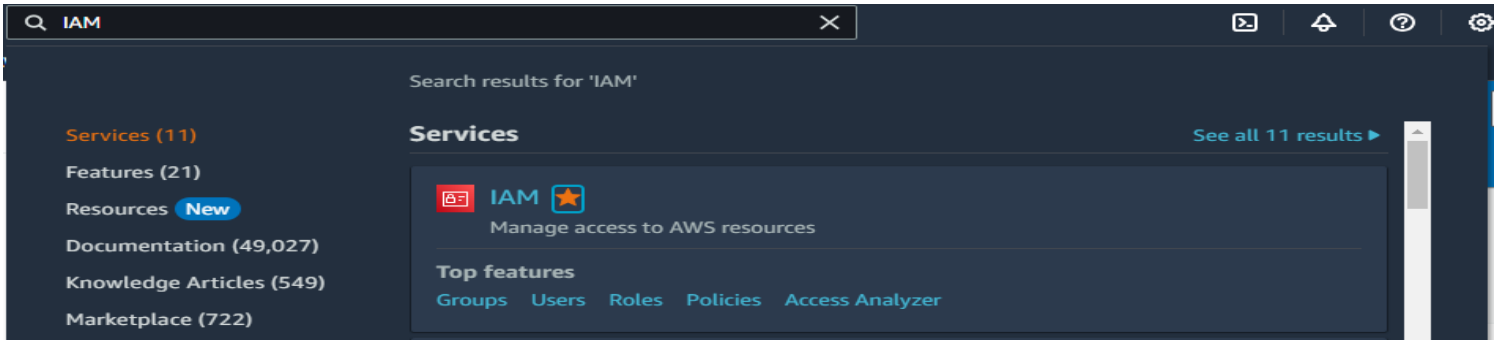


Now create IAM user, Give permission as a **AmazonDynamoDBReadOnlyAccess** and try to change in Dynamodb table. so it should have to give gives error as **permission denied .** let's check it out

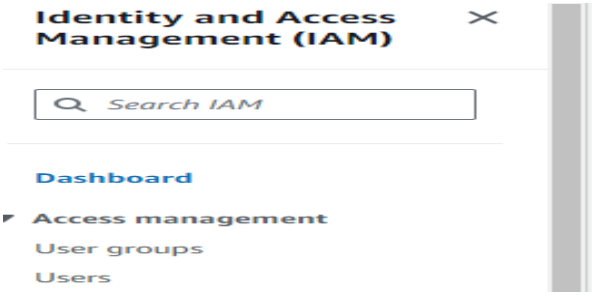
⭐ Create IAM User and Attach Policy

1. **Go the IAM :**

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

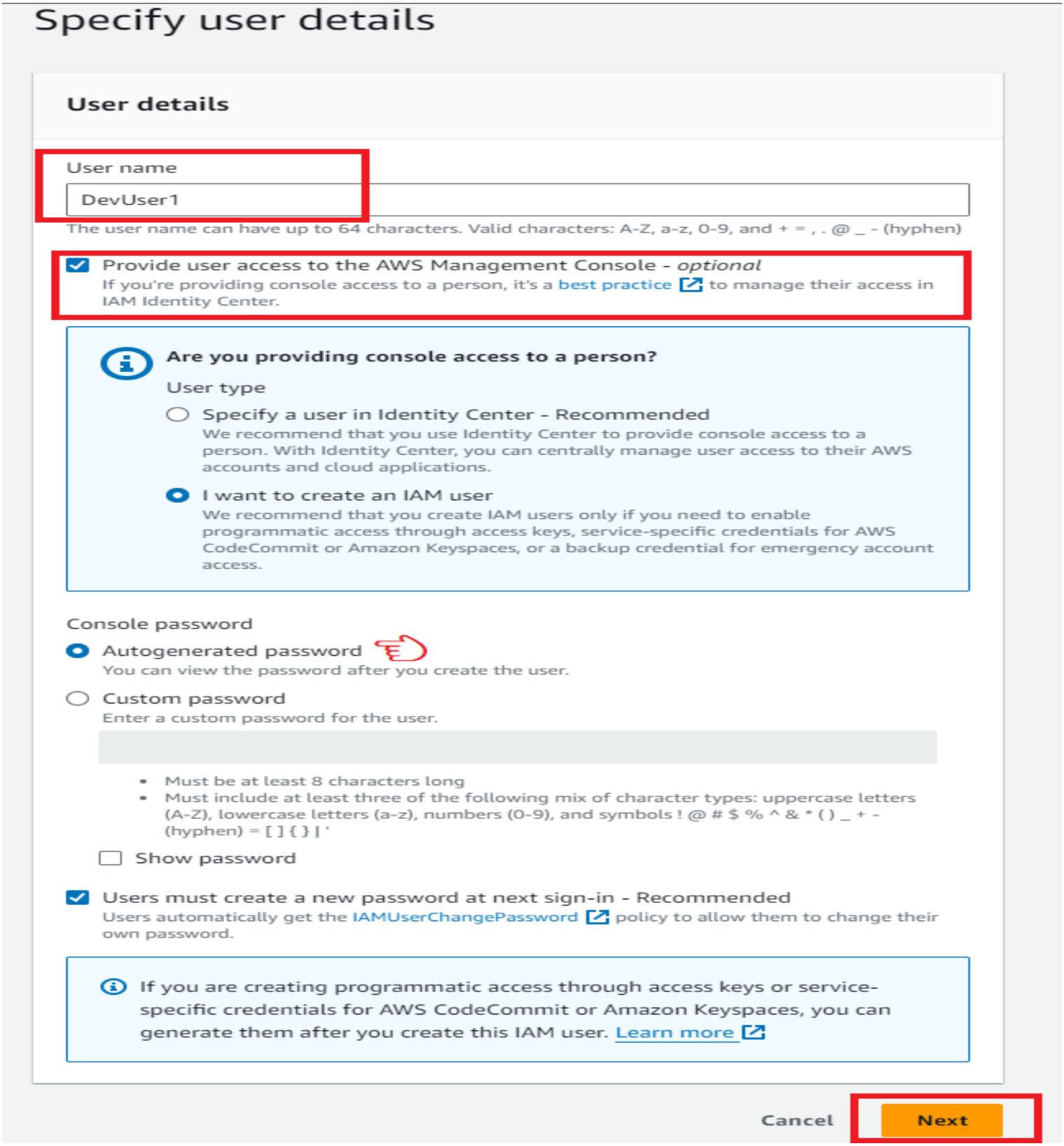


1. **Keep going to "Users"**: Select "Users" from the left navigation pane of the IAM console. Click "Add user".

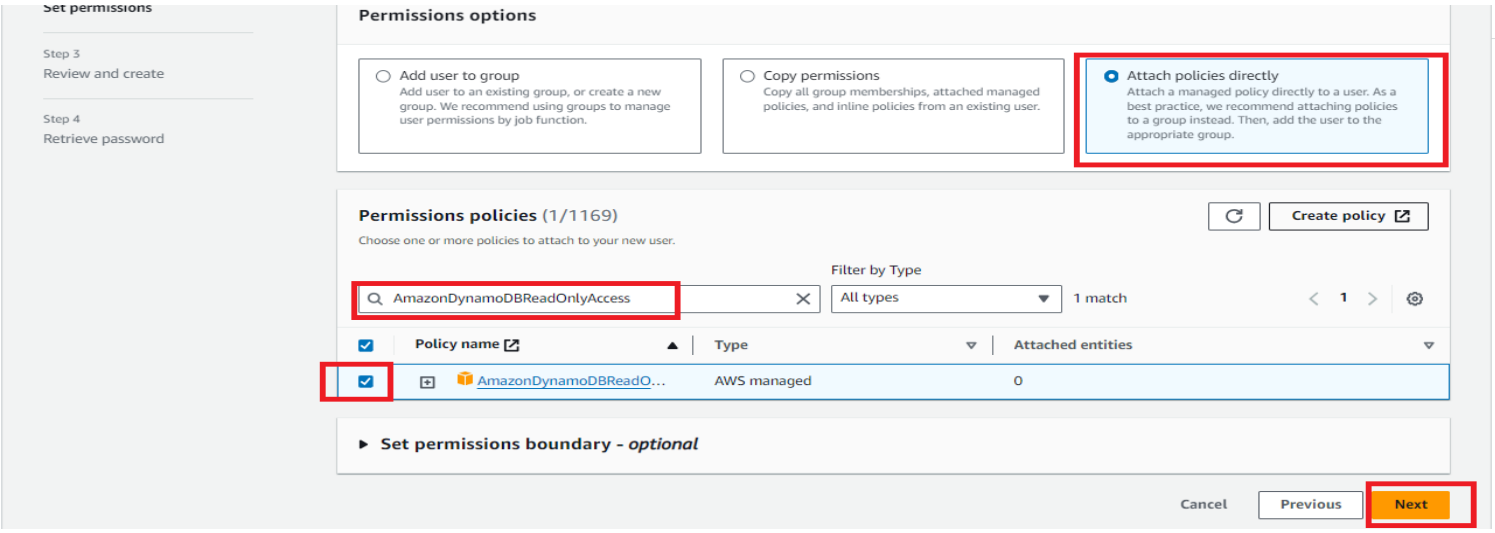


2. Click "**Create user**" from the menu. Enter the user's data.
3. Give the **IAM** user a **username**.Select the kind of access. Choose "Programmatic access" (for AWS CLI, SDK, etc.) for this scenario.If you would like to **grant console access**, you can select "AWS Management Console access" as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

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

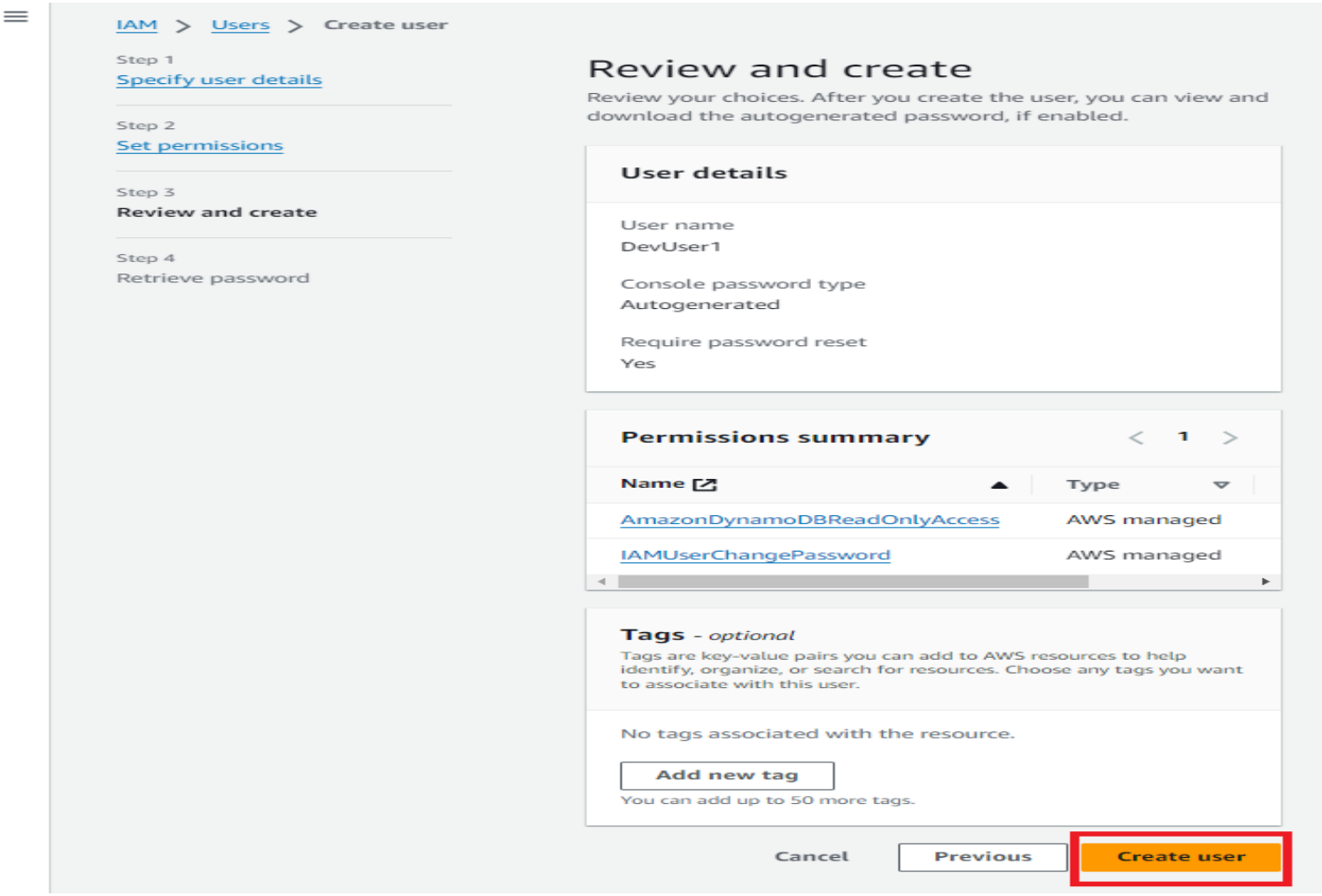


4. **Configure Permissions:**Go to "Set permissions" and select "Attach existing policies directly." Look for the "**AmazonDynamoDBReadOnlyAccess**" policy and attach it. DynamoDB can be accessed read-only with this policy.



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

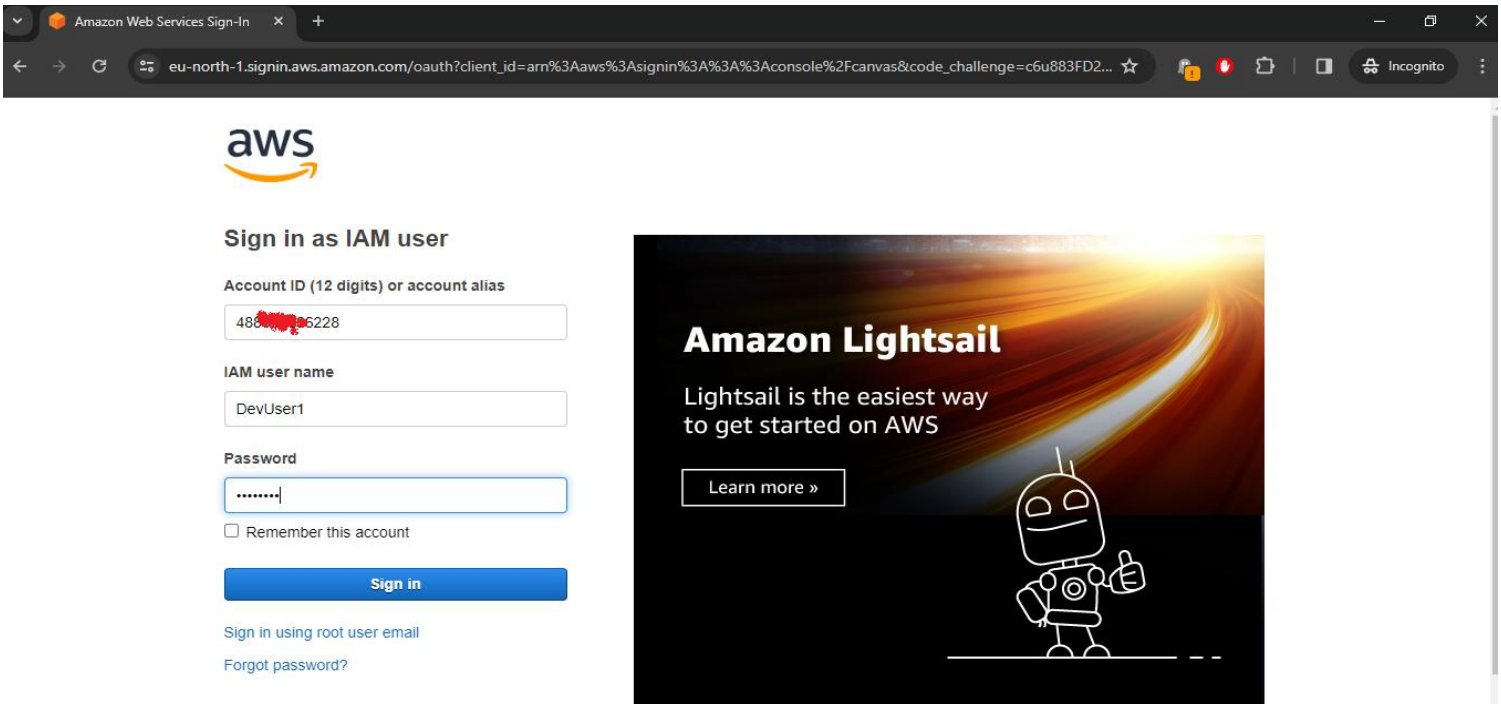
Create User: Select "Create user" from the menu.



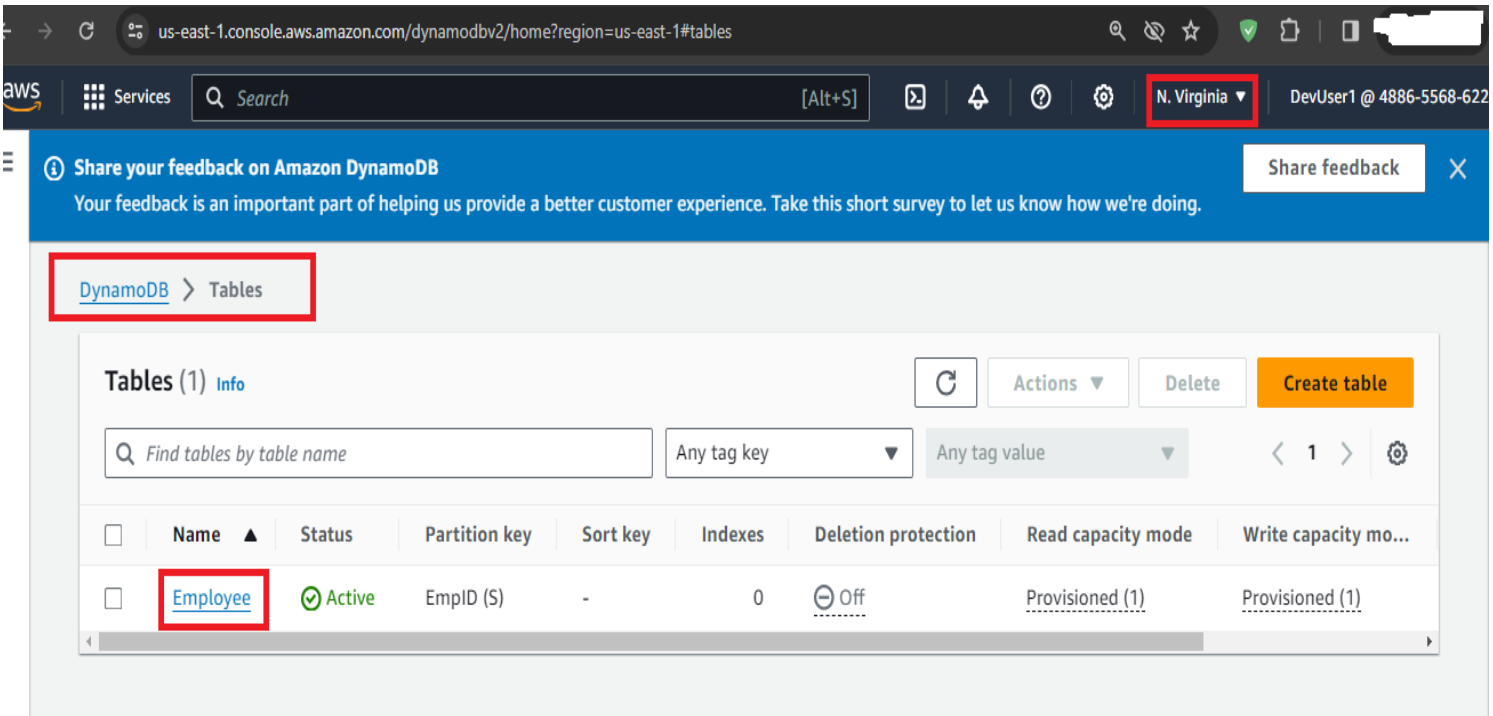
6. **Retrieve password :**The user's password can be viewed and downloaded below, and you can also send the user login credentials to access the **AWS Management Console**. You can only view and download this password at this time.

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

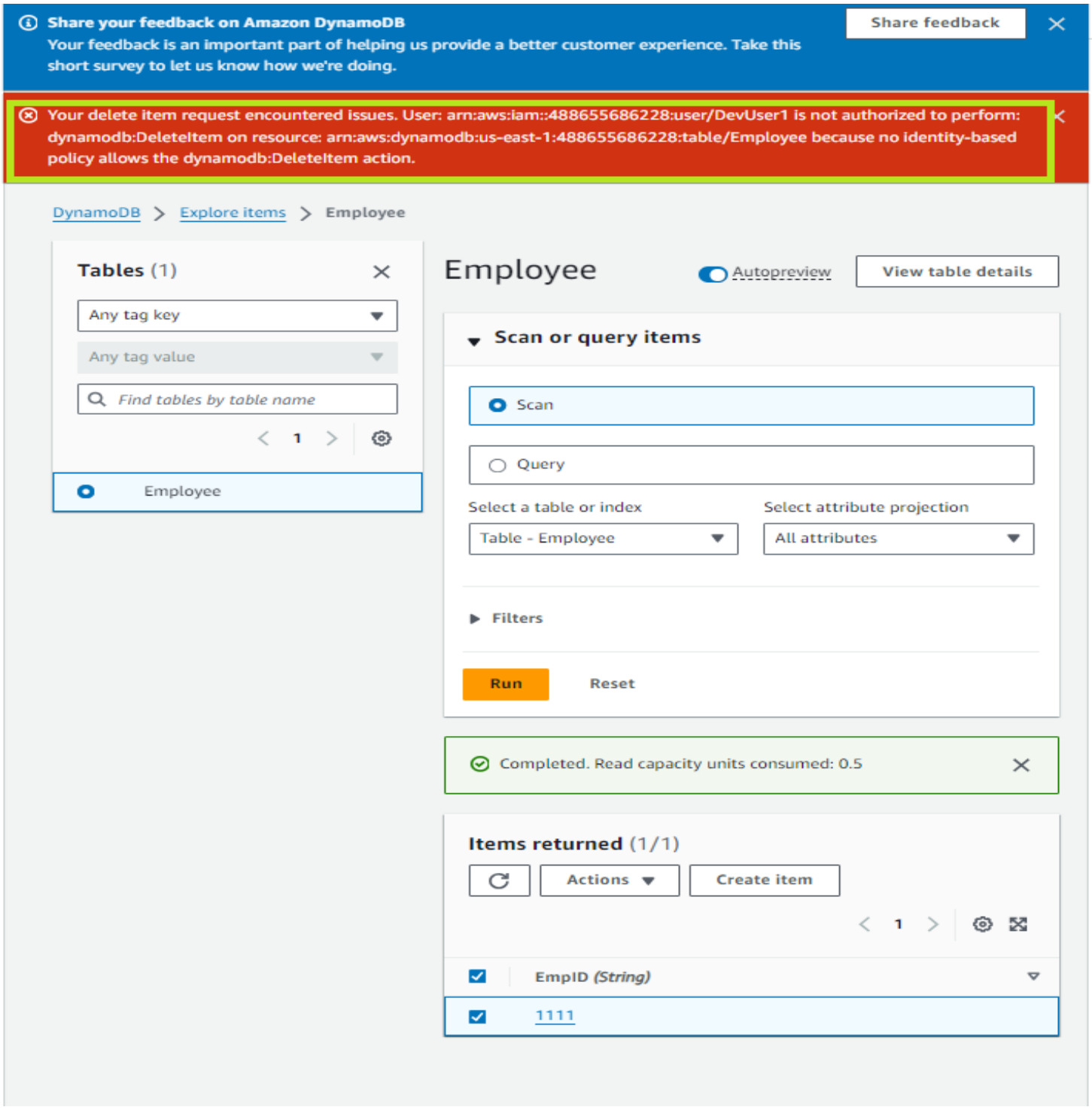
(In that file: Console sign-in URL, User name, Password using you can login & **check that you are able to change the data in Dynamodb table. let's do it.**



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



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



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!